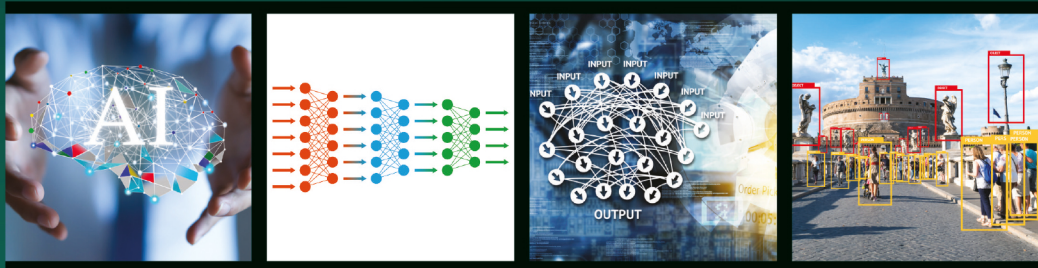


Компьютерное зрение

Передовые методы и глубокое обучение



Рой Дэвис • Мэтью Тёрк

Кэвин П. Мэрфи

**Компьютерное зрение.
Современные методы
и перспективы развития**

Advanced Methods and Deep Learning in Computer Vision

Edited by

E.R. Davies

Matthew A. Turk



ACADEMIC PRESS

An imprint of Elsevier

Компьютерное зрение. Современные методы и перспективы развития

Редакторы издания

Рой Дэвис
Мэтью Терк



Москва, 2022

УДК 004.8
ББК 32.81
К63

К63 Компьютерное зрение. Современные методы и перспективы развития / ред. Р. Дэвис, М. Терк; пер. с англ. В. С. Яценкова. – М.: ДМК Пресс, 2022. – 690 с.: ил.

ISBN 978-5-93700-148-1

Эта книга рассказывает о передовых методах компьютерного зрения. Показано, как искусственный интеллект обнаруживает признаки и объекты, на каких данных он обучается, на чем основано распознавание лиц и действий, отслеживание аномалий. Особое внимание уделяется методам глубокого обучения. Все ключевые принципы проиллюстрированы примерами из реальной практики.

Книга адресована исследователям и практикам в области передовых методов компьютерного зрения, а также тем, кто изучает эту технологию самостоятельно или в рамках вузовского курса.

УДК 004.8
ББК 32.81

This Russian edition of *Advanced Methods and Deep Learning in Computer Vision* (9780128221099) by E.R. Davies and Matthew Turk is published by arrangement with Elsevier Inc.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-0-12-822109-9 (англ.)
ISBN 978-5-93700-148-1 (рус.)

© Elsevier Inc., 2022
© Перевод, оформление, издание,
ДМК Пресс, 2022

Посвящаю эту книгу моей семье.

Светлой памяти моих родителей, Артура и Мэри Дэвис.

Моей жене Джоан за любовь, терпение, поддержку и вдохновение.

Моим детям Элизабет, Саре и Марион и внукам Джасперу, Джерому, Еве,
Таре и Пиа за то, что принесли бесконечную радость в мою жизнь!

— *Рой Дэвис*

Эта книга посвящается студентам, коллегам, друзьям и членам семьи,
которые мотивировали, направляли и поддерживали меня разными
способами, – всех невозможно перечислить поименно.

Моей жене Келли и детям Ханне и Мэтту – особая благодарность
и признательность за вашу любовь и вдохновение.

— *Мэтью Терк*

Содержание

От издательства.....	17
Список соавторов	18
О редакторах	20
Предисловие	21
Глава 1. Кардинальные перемены в области компьютерного зрения.....	27
1.1. Введение. Компьютерное зрение и его история	27
1.2. Часть А. Обзор операторов низкоуровневой обработки изображений	31
1.2.1. Основы обнаружения краев	31
1.2.2. Оператор Кэнни	33
1.2.3. Обнаружение сегмента линии	34
1.2.4. Оптимизация чувствительности обнаружения	35
1.2.5. Работа с изменениями интенсивности фона	37
1.2.6. Теория, сочетающая согласованный фильтр и конструкции с нулевым средним	37
1.2.7. Структура маски (дополнительные соображения).....	38
1.2.8. Обнаружение угла	40
1.2.9. Оператор «особой точки» Харриса	41
1.3. Часть В. Локализация и распознавание двухмерных объектов	43
1.3.1. Подход к анализу формы на основе центроидного профиля	43
1.3.2. Схемы обнаружения объектов на основе преобразования Хафа	46
1.3.3. Применение преобразования Хафа для обнаружения линий	50
1.3.4. Использование RANSAC для обнаружения линий	51
1.3.5. Теоретико-графовый подход к определению положения объекта	54
1.3.6. Использование обобщенного преобразования Хафа для экономии вычислений	57
1.3.7. Подходы на основе частей	59
1.4. Часть С. Расположение трехмерных объектов и важность неизменности	60
1.4.1. Введение в трехмерное зрение.....	60
1.4.2. Неоднозначность положения при перспективной проекции.....	64
1.4.3. Инварианты как помощь в трехмерном распознавании	68
1.4.4. Кросс-коэффициенты: концепция «отношения коэффициентов»	69
1.4.5. Инварианты для неколлинеарных точек	71
1.4.6. Обнаружение точки схода	73
1.4.7. Подробнее о точках схода	75
1.4.8. Промежуточный итог: значение инвариантов	76
1.4.9. Преобразование изображения для калибровки камеры	77
1.4.10. Калибровка камеры.....	80

1.4.11. Внутренние и внешние параметры	82
1.4.12. Многоракурсное зрение	83
1.4.13. Обобщенная геометрия стереозрения	84
1.4.14. Существенная матрица	85
1.4.15. Фундаментальная матрица	87
1.4.16. Свойства существенной и фундаментальной матриц	88
1.4.17. Расчет фундаментальной матрицы	88
1.4.18. Усовершенствованные методы триангуляции	89
1.4.19. Достижения и ограничения многоракурсного зрения	90
1.5. Часть D. Отслеживание движущихся объектов	90
1.5.1. Основные принципы отслеживания	90
1.5.2. Альтернативы вычитанию фона	94
1.6. Часть E. Анализ текстур	98
1.6.1. Введение	98
1.6.2. Основные подходы к анализу текстур	99
1.6.3. Метод Лоуза на основе энергии текстуры	101
1.6.4. Метод собственного фильтра Аде	103
1.6.5. Сравнение методов Лоуза и Аде	105
1.6.6. Последние разработки	106
1.7. Часть F. От искусственных нейронных сетей к методам глубокого обучения	106
1.7.1. Введение: как ИНС превратились в СНС	106
1.7.2. Параметры, определяющие архитектуру CNN	109
1.7.3. Архитектура сети AlexNet	110
1.7.4. Архитектура сети VGGNet Симоняна и Зиссермана	113
1.7.5. Архитектура DeconvNet	116
1.7.6. Архитектура SegNet	118
1.7.7. Применение глубокого обучения для отслеживания объектов	120
1.7.8. Применение глубокого обучения в классификации текстур	124
1.7.9. Анализ текстур в мире глубокого обучения	128
1.8. Часть G. Заключение	129
Благодарности	130
Литературные источники	130
Об авторе главы	135

Глава 2. Современные методы робастного обнаружения объектов

2.1. Введение	137
2.2. Предварительные положения	139
2.3. R-CNN	141
2.3.1. Внутреннее устройство	141
2.3.2. Обучение	142
2.4. Сеть SPP-Net	142
2.5. Сеть Fast R-CNN	143
2.5.1. Архитектура	144
2.5.2. Пулинг ROI	144

2.5.3. Многозадачная функция потери	145
Классификация	145
Регрессия ограничивающей рамки	145
2.5.4. Стратегия тонкой настройки	146
2.6. Faster R-CNN.....	146
2.6.1. Архитектура.....	147
2.6.2. Сети прогнозирования регионов	147
2.7. Каскадная R-CNN.....	149
2.7.1. Каскадная архитектура R-CNN.....	150
2.7.2. Каскадная регрессия ограничивающей рамки.....	151
2.7.3. Каскадное обнаружение.....	152
2.8. Представление разномасштабных признаков.....	152
2.8.1. Архитектура MC-CNN.....	154
2.8.1.1. Архитектура.....	154
2.8.2. Сеть FPN	155
2.8.2.1. Архитектура.....	156
2.9. Архитектура YOLO	158
2.10. Сеть SSD	159
2.10.1. Архитектура.....	159
2.10.2. Обучение	160
2.11. RetinaNet.....	161
2.11.1. Фокальная потеря.....	161
2.12. Производительность детекторов объектов	162
2.13. Заключение	163
Литературные источники	164
Об авторах главы.....	165

Глава 3. Обучение с ограниченным подкреплением – статические и динамические задачи

3.1. Введение.....	168
3.2. Контекстно-зависимое активное обучение	168
3.2.1. Активное обучение.....	169
3.2.2. Важность контекста активного обучения	172
3.2.3. Фреймворк контекстно-зависимого активного обучения	174
3.2.4. Практическое применение.....	177
3.3. Локализация событий при слабой разметке.....	180
3.3.1. Архитектура сети	183
3.3.2. k-матричное обучение.....	183
3.3.3. Сходство совместных действий.....	184
3.3.4. Практическая реализация	186
3.4. Семантическая сегментация с использованием слабой разметки	189
3.4.1. Слабые метки для классификации категорий.....	191
3.4.2. Слабые метки для выравнивания признаков.....	192
3.4.3. Оптимизация сети	194
3.4.4. Получение слабой разметки.....	195
3.4.5. Применения.....	196
3.4.6. Визуализация выходного пространства.....	198

3.5. Обучение с подкреплением со слабой разметкой для динамических задач.....	199
3.5.1. Обучение прогнозированию подцелей.....	202
3.5.2. Предварительное обучение с учителем	204
3.5.3. Практическое применение.....	204
3.6. Выводы.....	207
Благодарности	209
Литературные источники	209
Об авторах главы.....	215

Глава 4. Эффективные методы глубокого обучения..... 216

4.1. Сжатие модели.....	216
4.1.1. Прореживание параметров	217
4.1.2. Низкоранговая факторизация	220
4.1.3. Квантование	221
4.1.4. Дистилляция знаний.....	225
4.1.5. Автоматическое сжатие модели.....	226
4.2. Эффективные архитектуры нейронных сетей	230
4.2.1. Стандартный сверточный слой	231
4.2.2. Эффективные сверточные слои.....	231
4.2.3. Разработанные вручную эффективные модели CNN.....	232
4.2.4. Поиск нейронной архитектуры	236
4.2.5. Поиск нейронной архитектуры, ориентированной на оборудование	239
4.3. Заключение	246
Литературные источники	246

Глава 5. Условная генерация изображений и управляемая генерация визуальных паттернов..... 254

5.1. Введение	254
5.2. Изучение визуальных паттернов: краткий исторический обзор.....	258
5.3. Классические генеративные модели.....	260
5.4. Глубокие генеративные модели.....	261
5.5. Глубокая условная генерация изображений	266
5.6. Разделенные представления в управляемом синтезе паттернов	267
5.6.1. Разделение визуального содержания и стиля	267
5.6.2. Разделение структуры и стиля.....	274
5.6.3. Разделение личности и атрибутов	277
5.7. Заключение.....	284
Литературные источники	284

Глава 6. Глубокое распознавание лиц с использованием полных и частичных изображений..... 289

6.1. Введение	289
6.1.1. Модели глубокого обучения.....	291

6.2. Компоненты системы глубокого распознавания лиц	297
6.2.1. Пример обученной модели CNN для распознавания лиц	298
6.3. Распознавание лиц с использованием полных изображений лица	301
6.3.1. Проверка подобия с использованием модели FaceNet	303
6.4. Глубокое распознавание неполных изображений лица	304
6.5. Обучение специальной модели для полных и частичных изображений лица	307
6.5.1. Предлагаемая архитектура модели	309
6.5.2. Фаза обучения модели	309
6.6. Заключение	310
Литературные источники	312
Об авторе главы	313

Глава 7. Адаптация домена с использованием неглубоких и глубоких нейросетей, обучаемых без учителя

7.1. Введение	314
7.2. Адаптация домена с использованием многообразия	316
7.2.1. Адаптация домена без учителя с использованием произведения многообразий	317
7.3. Адаптация домена без учителя с использованием словарей	319
7.3.1. Общий словарь доменной адаптации	321
7.3.2. Совместная иерархическая адаптация домена и изучение признаков	325
7.3.3. Инкрементное изучение словаря для адаптации предметной области без учителя	330
7.4. Адаптация домена с использованием глубоких сетей, обучаемых без учителя	334
7.4.1. Дискриминационные подходы к адаптации предметной области	335
7.4.2. Генеративные подходы к адаптации домена	338
7.5. Заключение	346
Литературные источники	346
Об авторах главы	352

Глава 8. Адаптация домена и непрерывное обучение семантической сегментации

8.1. Введение	353
8.1.1. Формальная постановка задачи	355
8.2. Адаптация домена без учителя	356
8.2.1. Формулировка задачи адаптации домена	358
8.2.2. Основные подходы к адаптации	359
8.2.2.1. Адаптация на входном уровне	360
8.2.2.2. Адаптация на уровне признаков	361
8.2.2.3. Адаптация на уровне выхода	362
8.2.3. Методы адаптации домена без учителя	362
8.2.3.1. Состязательная адаптация домена	362
8.2.3.2. Генеративная адаптация	366

8.2.3.3. Несоответствие классификатора	368
8.2.3.4. Самостоятельное обучение	369
8.2.3.5. Многозадачность	372
8.3. Непрерывное обучение	373
8.3.1. Формулировка задачи непрерывного обучения	374
8.3.2. Особенности непрерывного обучения в семантической сегментации	376
8.3.3. Методы поэтапного обучения	378
8.3.3.1. Дистилляция знаний	378
8.3.3.2. Замораживание параметров	380
8.3.3.3. Геометрическая регуляризация на уровне признаков	380
8.3.3.4. Новые направления	381
8.4. Заключение	382
Благодарности	382
Литературные источники	382
Об авторах главы	389

Глава 9. Визуальное отслеживание движущихся объектов

9.1. Введение	390
9.1.1. Определение задачи отслеживания	390
9.1.2. Затруднения при отслеживании	391
9.1.3. Обоснование методики	392
9.1.4. Историческая справка	393
9.2. Методы на основе шаблонов	394
9.2.1. Основы	394
9.2.2. Показатели качества модели	396
9.2.3. Нормализованная кросс-корреляция	398
9.2.4. Чисто фазовый согласованный фильтр	399
9.3. Методы последовательного обучения	400
9.3.1. Фильтр MOSSE	401
9.3.2. Дискриминативные корреляционные фильтры	403
9.3.3. Подходящие признаки для DCF	405
9.3.4. Отслеживание в масштабном пространстве	406
9.3.5. Пространственное и временное взвешивание	408
9.4. Методы, основанные на глубоком обучении	410
9.4.1. Глубокие признаки в DCF	411
9.4.2. Адаптивные глубокие признаки	413
9.4.3. DCF сквозного обучения	414
9.5. Переход от отслеживания к сегментации	416
9.5.1. Сегментация видеообъектов	416
9.5.2. Генеративный метод VOS	417
9.5.3. Дискриминативный метод VOS	419
9.6. Выводы	420
Благодарности	421
Литературные источники	422
Об авторе главы	429

Глава 10. Длительное отслеживание объекта на основе глубокого обучения	430
10.1. Введение.....	431
10.1.1. Трудности отслеживания видеообъектов	432
10.1.1.1. Видовые проблемы отслеживания.....	432
10.1.1.2. Проблемы машинного обучения при отслеживании.....	433
10.1.1.3. Технические проблемы при отслеживании.....	435
10.2. Краткосрочное визуальное отслеживание объекта	435
10.2.1. Неглубокие трекеры	436
10.2.2. Глубокие трекеры.....	438
10.2.2.1. Отслеживание на основе корреляционного фильтра	438
10.2.2.2. Отслеживание на основе некорреляционных фильтров.....	440
10.3. Долгосрочное визуальное отслеживание объекта	441
10.3.1. Устаревание модели при длительном отслеживании	442
10.3.2. Исчезновение и повторное появление цели	446
10.3.3. Долгосрочные трекеры	446
10.3.3.1. Предварительное обучение и сиамские трекеры	446
10.3.4. Инвариантность и эквивариантность представления	452
10.3.4.1. Инвариантность при отслеживании.....	452
10.3.4.2. Эквивариантность при отслеживании	454
10.3.4.3. Эквивариантность переноса.....	456
10.3.4.4. Эквивариантность вращения	458
10.3.4.5. Эквивариантность масштаба.....	461
10.3.4.6. Эффективность сиамских трекеров.....	464
10.3.4.7. Гибридное обучение с сиамскими трекерами.....	464
10.3.4.8. Последовательное обучение помимо сиамских трекеров	467
10.3.5. Наборы данных и тесты	468
10.4. Заключение	468
Литературные источники	469
Об авторах главы.....	473

Глава 11. Обучение пониманию сцены на основании действий	474
11.1. Введение.....	474
11.2. Аффордансы объектов.....	476
11.2.1. Зачем аффордансы нужны компьютерному зрению?	477
11.2.2. Первые исследования на тему аффордансов.....	479
11.2.3. Обнаружение, классификация и сегментация аффордансов.....	480
11.2.3.1. Обнаружение аффордансов по геометрическим признакам	480
11.2.3.2. Семантическая сегментация и классификация по изображениям	482
11.2.4. Аффорданс в контексте распознавания действий и обучения роботов	484
11.2.4.1. Распознавание действий.....	484
11.2.4.2. Изучение аффордансов в зрении роботов.....	485

11.2.5. Промежуточный итог – изучение аффордансов	486
11.3. Функциональный анализ манипуляций	487
11.3.1. Активное взаимодействие между познанием и восприятием	487
11.3.2. Грамматика действий	488
11.3.2.1. Различные реализации грамматики	490
11.3.2.2. Являются ли грамматики выразительными и лаконичными описаниями?	491
11.3.3. Модули для понимания действий	491
11.3.3.1. Захватывание: важный признак для понимания действий	491
11.3.3.2. Геометрические факторы для робастизации	494
11.3.4. Проблематика понимания деятельности	495
11.4. Понимание функциональной сцены посредством глубокого обучения с помощью языка и зрения	496
11.4.1. Атрибуты в обучении без ознакомления	498
11.4.2. Общие пространства для встраивания	499
11.4.3. Построение семантических векторных пространств	502
11.4.3.1. word2vec	502
11.4.4. Общие пространства представления и графовые модели	503
11.5. Перспективные направления исследований	505
11.6. Выводы	507
Благодарности	508
Литературные источники	508
Об авторах главы	513

Глава 12. Сегментация событий во времени

с использованием когнитивного самообучения	515
12.1. Введение	516
12.2. Теория сегментации событий в когнитивной науке	518
12.3. Вариант 1: однопроходная сегментация во времени с использованием предсказания	521
12.3.1. Извлечение и кодирование признаков	523
12.3.2. Рекуррентное прогнозирование для прогнозирования признаков	524
12.3.3. Реконструкция признаков	525
12.3.4. Функция потерь при самообучении	525
12.3.5. Механизм стробирования на основе ошибок	526
12.3.6. Адаптивное обучение для повышения робастности	527
12.3.7. Промежуточный итог	529
12.3.7.1. Наборы данных	529
12.3.7.2. Метрики оценки	529
12.3.7.3. Вариативные исследования	530
12.3.7.4. Количественная оценка	531
12.3.7.5. Качественная оценка	533
12.4. Вариант 2: сегментация с использованием моделей событий на основе внимания	534
12.4.1. Извлечение признаков	536

12.4.2. Модуль внимания	537
12.4.3. Функция потерь, взвешенная по движению.....	537
12.4.4. Результаты	538
12.4.4.1. Набор данных.....	539
12.4.4.2. Критерии оценки.....	539
12.4.4.3. Вариативные исследования	540
12.4.4.4. Количественная оценка.....	542
12.4.4.5. Качественная оценка	542
12.5. Вариант 3: пространственно-временная локализация с использованием карты предсказательных потерь	544
12.5.1. Извлечение признаков.....	544
12.5.2. Иерархический стек предсказания	546
12.5.3. Потеря предсказания	547
12.5.4. Извлечение каналов действий.....	548
12.5.5. Результаты	548
12.5.5.1. Данные	548
12.5.5.2. Показатели и базовые уровни	549
12.5.5.3. Количественная оценка.....	550
12.5.5.4. Качественная оценка	554
12.6. Другие подходы к сегментации событий в компьютерном зрении.....	556
12.6.1. Методы на основе обучения с учителем	556
12.6.2. Методы на основе частичного обучения с учителем	557
12.6.3. Методы на основе обучения без учителя	557
12.6.4. Методы на основе самообучения	558
12.7. Выводы	559
Благодарности	560
Литературные источники	560
Об авторах главы.....	567

Глава 13. Вероятностные методы обнаружения аномалий в данных временных рядов с использованием обученных моделей для мультимедийных самосознательных систем

13.1. Введение	569
13.2. Базовые понятия и текущее положение дел	571
13.2.1. Генеративные модели	571
13.2.2. Модели динамической байесовской сети (DBN).....	571
13.2.3. Вариационный автокодировщик	573
13.2.4. Типы аномалий и методы обнаружения аномалий	574
13.2.5. Обнаружение аномалий в данных низкой размерности.....	577
13.2.6. Обнаружение аномалий в многомерных данных.....	578
13.3. Архитектура вычисления аномалии в самосознательных системах	579
13.3.1. Общее описание архитектуры	579
13.3.2. Модель обобщенной динамической байесовской сети (GDBN).....	581
13.3.3. Алгоритм логического вывода в реальном времени.....	584
13.3.4. Измерения мультимодальных аномалий	586
13.3.4.1. Дискретный уровень.....	588

13.3.4.2. Непрерывный уровень	588
13.3.4.3. Уровень наблюдения	589
13.3.5. Использование обобщенных ошибок для непрерывного обучения.....	589
13.4. Пример: обнаружение аномалий в мультисенсорных данных от автомобиля с самосознанием.....	590
13.4.1. Описание условий эксперимента.....	590
13.4.2. Обучение модели DBN	591
13.4.3. Многоуровневое обнаружение аномалий.....	592
13.4.3.1. Задача объезда пешеходов.....	593
13.4.3.2. Задача разворота	594
13.4.3.3. Аномалии на уровне изображения	596
13.4.3.4. Оценка обнаружения аномалий.....	596
13.4.4. Аномалии проприоцептивных сенсорных данных.....	598
13.4.5. Дополнительные результаты	599
13.5. Выводы.....	600
Литературные источники	600
Об авторах главы	603

Глава 14. Методы PnP и глубокой развертки

для восстановления изображения	605
14.1. Введение	605
14.2. Алгоритм полуквадратичного разделения (HQS)	609
14.3. Глубокое восстановление изображения по методу PnP	610
14.3.1. Предварительное изучение глубокого шумоподавителя CNN	612
14.3.1.1. Шумоподавляющая сетевая архитектура	613
14.3.2. Методика обучения	614
14.3.3. Результаты удаления шума	615
14.3.3.1. Удаление шума с изображений в градациях серого.....	615
14.3.3.2. Удаление шума с цветного изображения.....	616
14.3.4. Алгоритм HQS для PnP IR	617
14.3.4.1. Алгоритм полуквадратичного разделения (HQS).....	617
14.3.4.2. Общая методика настройки параметров.....	617
14.3.4.3. Периодический геометрический самосогласованный ансамбль	618
14.4. Восстановление изображения методом глубокой развертки.....	619
14.4.1. Сеть глубокой развертки.....	620
14.4.1.1. Модуль данных \mathcal{D}	620
14.4.1.2. Модуль приора \mathcal{P}	620
14.4.1.3. Модуль гиперпараметров \mathcal{H}	621
14.4.2. Сквозное обучение	622
14.5. Эксперименты	622
14.5.1. Устранение размытия изображения	623
14.5.1.1. Количественные и качественные результаты.....	624
14.5.1.3. Промежуточные результаты.....	625
14.5.2. Сверхразрешение одиночного изображения (SISR).....	627

14.5.2.1. Количественное и качественное сравнение.....	628
14.6. Заключение	632
Благодарности	633
Литературные источники	633
Об авторах главы.....	638

Глава 15. Атаки на визуальные системы и защита

от злоумышленников	640
15.1. Введение.....	640
15.2. Определение проблемы	641
15.3. Свойства состязательной атаки	643
15.4. Типы возмущений.....	644
15.5. Сценарии атаки	645
15.5.1. Целевые модели	645
15.5.1.1. Модели для задач, связанных с изображениями.....	648
15.5.1.2. Модели для видеозадач	649
15.5.2. Наборы данных и метки	651
15.5.2.1. Наборы данных изображений	651
15.5.2.2. Наборы видеоданных	652
15.6. Обработка изображений	654
15.7. Классификация изображений.....	655
15.7.1. Белый ящик, ограниченные атаки	655
15.7.2. Белый ящик, атаки на основе контента.....	659
15.7.3. Атаки методом черного ящика	659
15.8. Семантическая сегментация и обнаружение объектов	661
15.9. Отслеживание объекта	662
15.10. Классификация видео	664
15.11. Защита от состязательных атак противника	666
15.11.1. Обнаружение атаки	666
15.11.2. Маскировка градиента.....	668
15.11.3. Устойчивость модели	670
15.12. Выводы.....	672
Благодарность.....	673
Литературные источники	673
Об авторах главы.....	682

Предметный указатель.....	683
----------------------------------	------------

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Elsevier очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Список соавторов

Сатьянараяна Аакур, факультет информатики, Государственный университет Оклахомы, Стиллуотер, Оклахома, США

Йогеш Балахи, факультет информатики и UMACS, Мэрилендский университет, Колледж-Парк, Мэриленд, США

Хан Цай, Массачусетский технологический институт, Кембридж, Массачусетс, США

Чжаовой Цай, Amazon Web Services, Пасадена, Калифорния, США

Андреа Кавалларо, Центр интеллектуального восприятия, Лондонский университет Королевы Марии, Лондон, Соединенное Королевство

Рама Челлаппа, факультеты электроники, вычислительной техники и биомедицинской инженерии, Университет Джона Хопкинса, Балтимор, Мэриленд, США

Дондон Чен, Microsoft Cloud & AI, Редмонд, Вашингтон, США

Э. Р. Дэвис Ройал Холлоуэй, Лондонский университет, Эгам, графство Суррей, Соединенное Королевство

Михаэль Фельсберг, Лаборатория компьютерного зрения, факультет электроники, Линчепингский университет, Линчёпинг, Швеция; Инженерная школа Университета Квазулу-Натал, Дурбан, Южная Африка

Корнелия Фермюллер, Университет Мэриленда, Институт перспективных компьютерных исследований, Центр компьютерных наук и инженерии Ирибе, Колледж-Парк, Мэриленд, США

Эфстратиос Гаввес, Институт информатики при Амстердамском университете, Амстердам, Нидерланды

Дипак Гупта, Институт информатики при Амстердамском университете, Амстердам, Нидерланды

Сонг Хан, Массачусетский технологический институт, Кембридж, Массачусетс, США

Ганг Хуа, Wormpex AI Research, Белвью, Вашингтон, США

Али Краяни, DITEN, Генуэзский университет, Генуя, Италия

Цзи Линь, Массачусетский технологический институт, Кембридж, Массачусетс, США

Лучио Марсенаро, DITEN, Генуэзский университет, Генуя, Италия

Майкл Мейнорд, Университет Мэриленда, факультет компьютерных наук, Центр компьютерных наук и инженерии Ирибе, Колледж-Парк, Мэриленд, США

Умберто Микьели, кафедра информационных технологий, Университет Падуи, Падуя, Италия

Рами Мунир, кафедра вычислительной техники и технологии, Университет Южной Флориды, Тампа, Флорида, США

Хиен Нгуен, факультет электроники и вычислительной техники, Хьюстонский университет, Хьюстон, Техас, США

- Чанги О**, Центр интеллектуального восприятия, Лондонский университет Королевы Марии, Лондон, Соединенное Королевство
- Суджой Пол**, Google Research, Бангалор, Индия
- Карло Регаццони**, DITEN, Генуэзский университет, Генуя, Италия
- Амит Рой-Чоудхури**, факультет электроники и вычислительной техники, Калифорнийский университет, Риверсайд, Калифорния, США
- Судип Саркар**, кафедра компьютерных наук и технологии, Университет Южной Флориды, Тампа, Флорида, США
- Джулия Славик**, DITEN, Генуэзский университет, Генуя, Италия
- Раду Тимофте**, Лаборатория компьютерного зрения, ETH Zürich, Цюрих, Швейцария
- Марко Тольдо**, кафедра информационных технологий, Университет Падуи, Падуя, Италия
- Хасан Угайл**, Центр цифровой обработки визуальной информации, Университет Брэдфорда, Брэдфорд, Великобритания
- Нуно Васконселос**, Калифорнийский университет в Сан-Диего, факультет электроники и вычислительной техники, Сан-Диего, Калифорния, США
- Алессио Зомперо**, Центр интеллектуального восприятия, Лондонский университет Королевы Марии, Лондон, Соединенное Королевство
- Пьетро Зануттиг**, кафедра информационных технологий, Университет Падуи, Падуя, Италия
- Кай Чжан**, Лаборатория компьютерного зрения, ETH Zürich, Цюрих, Швейцария

О редакторах

Рой Дэвис – почетный профессор факультета машинного зрения в Роял Холлоуэй, Лондонский университет. Он работал над многими аспектами зрения, от обнаружения признаков и подавления шума до робастного сопоставления образов и реализации практических задач зрения в реальном времени. Область его интересов включает автоматизированный осмотр объектов, наблюдение, управление транспортными средствами и раскрытие преступлений. Он опубликовал более 200 статей и три книги: *Machine Vision: Theory, Algorithms, Practicalities* (1990 г.), *Electronics, Noise and Signal Recovery* (1993 г.) и *Image Processing for the Food Industry* (2000 г.); первая из них не теряет популярности на протяжении 25 лет, а в 2017 г. вышло ее значительно расширенное пятое издание под названием *Computer Vision: Principles, Algorithms, Applications, Learning*. Рой является членом IoP и IET, а также старейшим членом IEEE. Он входит в редакционные коллегии журналов *Pattern Recognition Letters*, *Real-Time Image Processing*, *Imaging Science and IET Image Processing*. Он получил степень доктора наук в Лондонском университете; в 2005 г. был удостоен титула почетного члена BMVA, а в 2008 г. стал лауреатом премии Международной ассоциации распознавания образов.

Мэтью Тёрк – президент Технологического института Toyota в Чикаго (TTIC) и почетный профессор Калифорнийского университета в Санта-Барбаре. Его исследовательские интересы охватывают компьютерное зрение и взаимодействие человека с компьютером, включая такие темы, как автономные транспортные средства, распознавание лиц и жестов, мультимодальное взаимодействие, компьютерная фотография, дополненная и виртуальная реальность и этика ИИ. Он был главным организатором или ведущим нескольких крупных конференций, включая конференцию IEEE по компьютерному зрению и распознаванию образов, мультимедийную конференцию ACM, конференцию IEEE по автоматическому распознаванию лиц и жестов, международную конференцию ACM по мультимодальному взаимодействию и Зимнюю конференцию IEEE по приложениям компьютерного зрения. Он получил несколько наград за лучшую исследовательскую работу, а также различные премии и награды ACM, IEEE, IAPR и почетную премию Фулбрайта-Nokia за 2011–2012 гг. в области информационных и коммуникационных технологий.

Предисловие

Миновало почти десятилетие с тех пор, как произошел прорыв в разработке и применении *глубоких нейронных сетей* (deep neural network, DNN), и их последующий прогресс можно почти без преувеличения назвать выдающимся. Правда, этому прогрессу значительно способствовало появление специального оборудования в виде мощных графических процессоров; кроме того, возникло понимание, что *сверточные нейронные сети* (convolutional neural network, CNN) составляют важнейшую архитектурную основу, в которую можно встроить такие функции, как ReLU, упаковку, полностью связанные слои, распаковку и обратную свертку. По сути, все эти подходы помогли вдохнуть реальную жизнь в глубокие нейросети и резко расширить возможности их использования, поэтому первоначальный почти экспоненциальный рост их использования сохранился на весь последующий период. Мало того, что мощь нейросетевых технологий была впечатляющей, их применение значительно расширилось: от первоначального акцента на быстрое определение местоположения объекта и сегментацию изображения – и даже семантическую сегментацию – до применений, относящихся к видео, а не просто к анализу статичного изображения.

Было бы неправильно утверждать, что все развитие компьютерного зрения с 2012 г. было связано исключительно с появлением DNN. Свою роль сыграли и другие важные методы, такие как обучение с подкреплением, обучение с переносом, самообучение, лингвистическое описание изображений, распространение меток и такие приложения, как обнаружение новизны и аномалий, раскрашивание и отслеживание изображений. Тем не менее многие из упомянутых методов и области их применения получили новые стимулы и были пересмотрены и улучшены благодаря чрезвычайно быстрому внедрению DNN.

В этой книге мы попытались оценить, какие изменения произошли в области компьютерного зрения за минувшее десятилетие, насыщенное драматическими переменами. Сейчас самое время задаться вопросом, где мы находимся сейчас и насколько прочна база глубокого нейронного и машинного обучения, на которую опирается современное компьютерное зрение. Было ли это продуманное последовательное движение или слепой отчаянный рывок вперед? Не упускаем ли мы важные возможности и можем ли мы заглядывать в будущее с уверенностью, что движемся в правильном направлении? Или это тот случай, когда каждый исследователь может придерживаться своей собственной точки зрения и обращать внимание только на то, что представляется необходимым для его прикладной области, и если это так, то не ускользает ли от нас что-то важное при столь ограниченном подходе?

На самом деле есть и другие фундаментальные вопросы, на которые нужно найти ответ. Например, это сложный вопрос о том, до какой степени возможности глубокой нейросети можно повышать за счет качества обучающих данных; этот вопрос, по-видимому, применим к любому альтернативному

подходу, основанному на машинном обучении, независимо от того, относится ли он к DNN. Вряд ли фундаментальные ограничения нейросети зависят от того, каким способом ее обучали – обучение с подкреплением, самообучение или что-то другое. И обратите внимание, что люди вряд ли являются примером того, что можно каким-либо образом избежать интенсивного обучения; их способность к обучению с переносом лишь подтверждает, насколько эффективным может быть процесс обучения.

В этой книге мы стремимся не только представить передовые методики и подходы в области компьютерного зрения, но и разъяснить основополагающие принципы; мы взяли на себя роль преподавателей и, прежде чем представить читателю самые последние достижения, хотим сформировать у него понимание общей картины. Поэтому *глава 1* посвящена основам компьютерного зрения. Она начинается с детального анализа ранних подходов к компьютерному зрению, включая обнаружение признаков, обнаружение объектов, трехмерное зрение и появление DNN; далее мы переходим к визуальному слежению за объектами, которое рассматривается как пример прикладной области, где решающую роль могут играть DNN. Эта глава самая длинная в книге, потому что мы должны пройти путь с нуля до современных достижений; кроме того, она готовит почву для понимания ключевых идей и методов, описанных выдающимися экспертами в остальных главах.

Как будет показано в *главе 1*, обнаружение объектов – одна из самых сложных задач компьютерного зрения. В частности, эффективная система должна преодолевать такие проблемы, как искажение масштаба, окклюзия, переменное освещение, сложный фон и все факторы изменчивости, связанные с миром природы. *Глава 2* описывает различные методы и подходы, на которых основаны последние достижения. К ним относятся *слияние видимых областей* (region-of-interest pooling), *многозадачные потери* (multitask losses), *сети для предложения регионов* (region proposal networks), *привязки* (anchors), *каскадное обнаружение и регрессия* (cascaded detection and regression), *многомасштабные представления признаков* (multiscale feature representations), *методы дополнения данных* (data augmentation techniques), *функции потерь* (loss functions) и многое другое.

В *главе 3* подчеркивается, что недавние успехи в области компьютерного зрения в значительной степени связаны с появлением огромных массивов тщательно размеченных данных, необходимых для обучения моделей. В ней рассматриваются методы, которые можно использовать для обучения моделей распознавания на основе таких данных, требующие ограниченной ручной обработки. Помимо уменьшения количества размеченных вручную данных, необходимых для обучения моделей распознавания, необходимо снизить уровень подкрепления с сильного на слабый, в то же время разрешая релевантные запросы от оракула. Дан обзор теоретических основ и экспериментальных результатов, которые помогают достичь этого.

В *главе 4* рассматриваются вычислительные проблемы глубоких нейронных сетей, которые затрудняют их развертывание на оборудовании с ограниченными ресурсами. В ней обсуждаются методы сжатия моделей и поиска нейронной архитектуры, ориентированной на оборудование, с целью повышения эффективности глубокого обучения, уменьшения размера и ускоре-

ния нейронных сетей. В главе показано, как использовать *отсечение коэффициентов* (parameter pruning) для удаления избыточных весов, *факторизацию низкого ранга* (low-rank factorization) для уменьшения сложности, *квантование весов* (weight quantization) для уменьшения точности весов и размера модели, а также *дистилляцию знаний* (knowledge distillation) для переноса знаний из «черного ящика» больших моделей в меньшие.

В главе 5 обсуждается, как *глубокие генеративные модели* (deep generative models) пытаются восстановить низкоразмерную структуру целевых визуальных моделей. В ней показано, как использовать глубокие генеративные модели для достижения более управляемого синтеза визуальных паттернов посредством условной генерации изображения. Ключом к достижению этой цели является «распутывание» визуального представления, когда предпринимаются попытки разделить различные управляющие факторы в скрытом пространстве встраивания. Представлены три тематических исследования по *переносу стиля* (style transfer), *визуально-языковой генерации* (vision-language generation) и *синтезу лица* (face synthesis), чтобы проиллюстрировать, как этого добиться в условиях обучения без подкрепления или при слабом подкреплении.

Глава 6 посвящена актуальной проблеме реального мира – *распознаванию лиц* (face recognition). В ней обсуждаются современные методы, основанные на глубоком обучении, которые можно применять даже к неполным изображениям лица. В главе показано: (а) как создаются необходимые архитектуры глубокого обучения; (b) как такие модели можно обучать и тестировать; (с) как можно использовать *точную настройку* (fine tuning) предварительно обученных сетей для определения эффективных сигналов распознавания с полными и частичными данными о лице; (d) какие успехи достигнуты за счет последних разработок в области глубокого обучения; (е) каковы текущие ограничения методов глубокого обучения, используемых для распознавания лиц. В главе также упомянуты некоторые из нерешенных проблем в этой области.

В *главе 7* обсуждается важнейший вопрос о том, как перенести обучение из одной области данных в другую. Сюда относятся методы, основанные на дифференциальной геометрии, *разреженном представлении* (sparse representation) и глубоких нейронных сетях. Они делятся на два широких класса – дискриминационные и генеративные подходы. Первые включают обучение модели классификатора с использованием дополнительных потерь, чтобы сделать исходное и целевое распределения признаков похожими. Вторые используют генеративную модель для выполнения адаптации предметной области (домена): обычно междоменная генеративная состязательная сеть обучается для сопоставления образцов из исходного домена с целевым, а модель классификатора обучается на преобразованных целевых изображениях. Такие подходы проверяются на задачах междоменного распознавания и семантической сегментации.

В *главе 8* мы возвращаемся к задаче адаптации предметной области в контексте семантической сегментации, когда глубокие сети испытывают потребность в огромном количестве размеченных данных для обучения. Глава начинается с обсуждения различных уровней, на которых может осуществ-

ляться адаптация, и стратегий их достижения. Затем рассматривается задача непрерывного обучения семантической сегментации. Хотя эта задача является относительно новой областью исследований, интерес к ней быстро растет, и уже представлено множество различных сценариев. Они подробно описаны вместе с подходами, необходимыми для их решения.

Вслед за главой 1 в главе 9 вновь подчеркивается важность визуального отслеживания как одной из основных классических проблем компьютерного зрения. Цель этой главы – дать обзор развития области, начиная с алгоритма Лукаса–Канаде и согласованных фильтров и заканчивая подходами, основанными на глубоком обучении, а также переходом к сегментации видео. Обзор ограничен целостными моделями для общего отслеживания в плоскости изображения, и особое внимание уделяется дискриминационным моделям, трекеру MOSSE (minimum output sum of squared errors, минимальная выходная сумма квадратов ошибок) и DCF (discriminative correlation filters, дискриминационные корреляционные фильтры).

Глава 10 развивает концепцию визуального отслеживания объектов еще на один шаг и концентрируется на долгосрочном отслеживании. Чтобы успешно справиться с этой задачей, отслеживание объектов должно решать серьезные проблемы, связанные с *распадом модели* (model decay), то есть с ухудшением качества модели из-за нарастающей погрешности, а также с исчезновением и появлением цели. Успех глубокого обучения оказал большое влияние на подходы к отслеживанию визуальных объектов, поскольку автономное обучение *сиамских трекеров* (Siamese tracker) помогает устранить распад модели. Однако, чтобы избежать возможности потери отслеживания в тех случаях, когда внешний вид цели значительно меняется, сиамские трекеры могут воспользоваться встроенными инвариантностями и эквивариантностями, допускающими вариации внешнего вида, не усугубляя распад модели.

В последние годы крепнет уверенность в том, что в динамичной среде видео и движущихся объектов – особенно когда идет речь о *распознавании действий и поведения* (action/behavior recognition) – жизненно важную роль играет понимание когнитивных функций человека. Обоснованность этого предположения полностью подтверждают следующие две главы. В главе 11 описывается ориентированная на действия структура, которая охватывает несколько временных масштабов и уровней абстракции. Нижний уровень детализирует *характеристики объекта*, который совершает различные действия; средний уровень моделирует *отдельные действия*, а самый высокий уровень моделирует *деятельность*. Упор на использование характеристик понимания, геометрии, онтологий и ограничений, основанных на физике, позволяет избежать чрезмерного обучения характеристикам внешнего вида. Чтобы объединить восприятие на основе сигналов с *символьными знаниями* (symbolic knowledge), векторизованные знания согласовываются с визуальными признаками. Глава также включает обсуждение понятий *действия* (action) и *деятельности* (activity).

В главе 12 рассматривается проблема *временной сегментации событий* (temporal event segmentation). Достижения когнитивной науки демонстрируют подходы к разработке высокоэффективных алгоритмов компьютерного зрения для пространственно-временной сегментации событий в видео

без необходимости использования каких-либо аннотированных данных. Во-первых, модель теории сегментации событий позволяет вычислять границы событий: затем следует временная сегментация с использованием фреймворка прогнозирования восприятия, временная сегментация вместе с рабочими моделями событий, основанными на *картах внимания* (attention map), и пространственно-временная локализация событий. Этот подход обеспечивает производительность на современном уровне при временной сегментации без подкрепления и пространственно-временной локализации действий, позволяя конкурировать с производительностью базовых моделей, обучаемых с полным подкреплением и требующих большого объема полностью аннотированных данных.

Методы обнаружения аномалий лежат в основе многих приложений, таких как анализ медицинских изображений, обнаружение мошенничества или видеонаблюдение. Эти методы также представляют собой важный шаг на пути развития искусственных *саморазвивающихся систем* (self-aware system), которые могут постоянно учиться в новых ситуациях. В *главе 13* представлен метод обнаружения аномалий с частичным подкреплением для этого типа саморазвивающихся агентов. Он использует возможности передачи сообщений в обобщенных динамических байесовских сетях для выявления аномалий на разных уровнях абстракции и различных типов данных временных рядов. Следовательно, обнаруженные аномалии могут быть использованы для обеспечения саморазвития системы за счет интеграции новых приобретенных знаний. В главе рассмотрено исследование по тематике обнаружения аномалий с использованием мультисенсорных данных от полуавтономного транспортного средства, которое выполняет различные задачи в закрытой среде.

Методы, основанные на моделировании и машинном обучении, отражали две доминирующие стратегии при решении различных проблем восстановления изображений, когда речь идет о низкоуровневом техническом зрении. Как правило, эти два метода имеют свои достоинства и недостатки; например, методы на основе моделей обладают гибкостью при решении различных проблем восстановления изображений, но обычно требуют долгой и трудоемкой настройки априорных значений для обеспечения хорошей производительности. С другой стороны, методы на основе машинного обучения демонстрируют более высокую эффективность и результативность по сравнению с традиционными методами на основе моделей, в основном из-за сквозного обучения, но, как правило, им не хватает гибкости для решения различных задач восстановления изображений. *Глава 14* знакомит читателей с методами plug-and-play и развертывания на базе глубоких нейросетей, которые продемонстрировали большие перспективы за счет использования как методов, основанных на обучении, так и методов, основанных на модели: основная идея методов глубокого plug-and-play заключается в том, что шумоподаватель на основе машинного обучения может неявно служить исходным изображением для методов восстановления изображений на основе модели, в то время как идея методов глубокого развертывания заключается в том, что путем развертывания моделей с помощью переменных алгоритмов разделения можно получить сквозную обучаемую итеративную

сеть, заменяя соответствующие подзадачи нейронными модулями. Следовательно, методы глубокого plug-and-play и глубокого развертывания могут унаследовать гибкость методов, основанных на моделях, сохраняя при этом преимущества методов, основанных на обучении.

Визуальные состязательные объекты (visual adversarial examples) – это изображения и видео, намеренно искаженные, чтобы ввести в заблуждение модели машинного обучения. В *главе 15* представлен обзор методов формирования помех для создания визуальных состязательных объектов, применяемых при оценке решения задач классификации изображений, обнаружения объектов, отслеживания движения и распознавания видео. Сначала определяются ключевые свойства состязательной атаки и типы возмущений, порождаемых атакой; затем анализируются основные варианты методов генерации состязательных атак на изображения и видео и исследуются применяемые при этом знания о целевой модели. Наконец, рассмотрены защитные механизмы, которые повышают устойчивость моделей машинного обучения к атакам со стороны противника и вероятность выявления манипуляций входными данными.

Вместе эти главы раскрывают заинтересованному читателю – будь то студент, исследователь или практический специалист – всю ширину и глубину современной методологии компьютерного зрения.

В заключение мы хотели бы выразить всем авторам нашу благодарность за огромный энтузиазм и самоотверженность, проявленные при работе над главами этой монографии. Благодаря их усилиям эта книга, как мы надеемся, станет надежным путеводителем в мире современного компьютерного зрения. Благодарим Тима Питтса из Elsevier Science за его постоянные советы и поддержку с самого начала и на протяжении всего времени, пока мы трудились над составлением этого сборника.

Рой Дэвис

Роял Холлоуэй, Лондонский университет,
Лондон, Соединенное Королевство

Мэтью Терк

Технологический институт Toyota в Чикаго,
Чикаго, Иллинойс, США
Май 2021 г.

Глава 1

Кардинальные переменны в области компьютерного зрения

*Автор главы: Рой Дэвис,
Роял Холлоуэй, Лондонский университет, Эгам,
графство Суррей, Соединенное Королевство*

Краткое содержание главы:

- обзор истории методов компьютерного зрения, включая операторы низкогоуровневой обработки изображений, обнаружение 2D- и 3D-объектов, определение местоположения и распознавание, отслеживание и сегментацию;
- изучение развития методов глубокого обучения на основе искусственных нейронных сетей, включая взрывной рост популярности глубокого обучения;
- обзор методов глубокого обучения, применяемых для обнаружения признаков, обнаружения объектов, определения местоположения, распознавания и отслеживания объектов, классификации текстур и семантической сегментации изображений;
- влияние методов глубокого обучения на традиционную методологию компьютерного зрения.

1.1. ВВЕДЕНИЕ. КОМПЬЮТЕРНОЕ ЗРЕНИЕ И ЕГО ИСТОРИЯ

В течение последних трех-четырёх десятилетий компьютерное зрение постепенно превратилось в полноценный научный предмет со своей методологией и областью применения. На самом деле у него так много областей применения, что трудно перечислить их все. Среди наиболее известных – распознавание объектов, наблюдение (включая подсчет людей и распозна-

вание номерных знаков), роботизированное управление (включая автоматическое управление транспортным средством), сегментация и интерпретация медицинских изображений, автоматический осмотр и сборка в заводских условиях, распознавание отпечатков пальцев и лиц, интерпретация жестов и многое другое. Для работы компьютерного зрения необходим поток данных из различных источников изображений, включая каналы видимого и инфракрасного спектра, трехмерные датчики и ряд жизненно важных медицинских устройств визуализации, таких как компьютерные и магнитно-резонансные томографы. К тому же данные должны включать положение, позу, расстояние между объектами, движение, форму, текстуру, цвет и многие другие аспекты. При таком разнообразии данных и изобилии действий и методов, используемых для их обработки, будет трудно обрисовать общую картину в рамках одной главы: следовательно, выбор материала неизбежно будет ограничен; тем не менее мы будем стремиться обеспечить прочную основу и дидактический подход к предмету.

Сегодня вряд ли можно представить компьютерное зрение без огромного прорыва, достигнутого в 2010-х годах, и, в частности, «взрыва глубокого обучения», который произошел примерно в 2012 г. Это событие значительно изменило саму суть предмета исследований и привело к достижениям и применениям, которые не только впечатляют, но и во многих случаях выходят далеко за рамки того, о чем люди мечтали даже в 2010 г. Наша книга в первую очередь посвящена самым передовым достижениям в области компьютерного зрения; роль этой вступительной главы состоит в том, чтобы обрисовать в общих чертах историю традиционной методологии, исследовать новые методы глубокого обучения и показать, как они изменили и улучшили более ранние (устаревшие) подходы.

На первом этапе будет полезно рассмотреть истоки компьютерного зрения, которое можно считать зародившимся в 1960-х и 1970-х гг., в основном как ответвление обработки изображений. В то время появилась техническая возможность захватывать целые изображения, а также удобно хранить и обрабатывать их на цифровых компьютерах. Первоначально изображения, как правило, записывались в бинарном виде или в оттенках серого, хотя позже стало возможным захватывать их в цвете. Исследователи уже тогда мечтали подражать человеческому глазу, распознавая объекты и интерпретируя сцены, но с доступными тогда маломощными компьютерами эти мечты были далеки от воплощения. На практике обработка изображений использовалась для *исправления* (tidying up) изображений и обнаружения признаков объектов, а распознавание изображений осуществлялось с использованием методов статистического распознавания образов, таких как *алгоритм ближайшего соседа* (nearest neighbor algorithm). Двумя основными локомотивами развития компьютерного зрения стали искусственный интеллект и биологическое зрение. Ограниченный объем книги не позволит нам здесь обсуждать эти аспекты; отметим лишь, что они заложили основу искусственных нейронных сетей и глубокого обучения (подробнее об этом в разделе 1.7).

Исправление изображений, вероятно, лучше описать как предварительную обработку: она может включать в себя ряд функций, где одной из самых важных является устранение шума. Вскоре было обнаружено, что использо-

вание алгоритмов сглаживания, в которых вычисляется среднее значение интенсивностей в окне вокруг каждого входного пикселя, применяемое для формирования отдельного сглаженного изображения, не только приводит к снижению уровня шума, но и к влиянию сигнала на самого себя (этот процесс также можно представить как уменьшение входной полосы пропускания для устранения большей части шума, с дополнительным эффектом устранения из входного сигнала компонентов высокой пространственной частоты). Однако эта проблема была в значительной степени решена за счет применения медианной, а не средней фильтрации, поскольку она работает за счет устранения выбросов на каждом конце локального распределения интенсивности – медиана является значением, наименее подверженным влиянию шума.

Типичные ядра фильтрации по среднему показаны ниже, причем второе из них более приближено к идеальной гауссовой форме:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (1.1)$$

Оба они являются ядрами линейной свертки, которые по определению пространственно инвариантны в пространстве изображений. Общая маска свертки 3×3 задается выражением

$$\begin{bmatrix} c4 & c3 & c2 \\ c5 & c0 & c1 \\ c6 & c7 & c8 \end{bmatrix}, \quad (1.2)$$

где локальным пикселям присвоены метки 0–8. Затем мы берем значения интенсивности в локальной окрестности изображения 3×3 как

$$\begin{bmatrix} P4 & P3 & P2 \\ P5 & P0 & P1 \\ P6 & P7 & P8 \end{bmatrix}. \quad (1.3)$$

Воспользовавшись нотацией условного языка программирования наподобие C++, мы можем записать полную процедуру свертки в виде псевдокода:

$$\begin{aligned} &\text{для всех пикселей изображения выполнить } \{ \\ &\quad Q0 = P0*c0 + P1*c1 + P2*c2 + P3*c3 + P4*c4 \\ &\quad \quad + P5*c5 + P6*c6 + P7*c7 + P8*c8; \\ &\} \end{aligned} \quad (1.4)$$

До сих пор мы рассматривали маски свертки, которые представляют собой линейные комбинации входных интенсивностей: они отличаются от нелинейных процедур, таких как пороговая обработка, которые не могут быть выражены как свертки. На самом деле пороговая обработка очень широко применяется и может быть записана в виде следующего алгоритма:

```

для всех пикселей изображения выполнить {
    если ( $P_0 < \text{порог}$ )  $A_0 = 1$ ; иначе  $A_0 = 0$ ;
}

```

(1.5)

Эта процедура преобразует изображение в оттенках серого в Р-пространстве в бинарное изображение в А-пространстве. Здесь она используется для выделения темных объектов, представляя их как единицы на фоне нулей.

Мы завершаем этот раздел полной процедурой медианной фильтрации в пределах окрестности 3×3 :

```

для ( $i = 0$ ;  $i \leq 255$ ;  $i++$ )  $\text{hist}[i] = 0$ ;
для всех пикселей изображения выполнить {
    для ( $m = 0$ ;  $m \leq 8$ ;  $m++$ )  $\text{hist}[P[m]]++$ ;
     $i = 0$ ;  $\text{sum} = 0$ ;
    пока ( $\text{sum} < 5$ ) {
         $\text{sum} = \text{sum} + \text{hist}[i]$ ;
         $i = i + 1$ ;
    }
     $Q_0 = i - 1$ ;
    для ( $m = 0$ ;  $m \leq 8$ ;  $m++$ )  $\text{hist}[P[m]] = 0$ ;
}

```

(1.6)

Запись $P[0]$ обозначает P_0 , и так далее от $P[1]$ до $P[8]$. Заметим, что операция нахождения медианы требует больших вычислений, поэтому время экономится только за счет повторной инициализации конкретных элементов гистограммы, которые фактически использовались.

Важная особенность процедур, описываемых уравнениями (1.4)–(1.6), заключается в том, что они берут входные данные из одного пространства изображений и выводят их в другое пространство изображений – процесс, часто описываемый как параллельная обработка, – тем самым устраняя проблемы, связанные с порядком, в котором выполняются вычисления отдельных пикселей.

Наконец, все алгоритмы сглаживания изображений, задаваемые уравнениями (1.1)–(1.4), используют ядра свертки 3×3 , хотя, очевидно, можно использовать ядра гораздо большего размера: действительно, их можно реализовать иным путем, сначала преобразовывая в область пространственных частот, а затем систематически устраняя высокие пространственные частоты, хотя и с дополнительной вычислительной нагрузкой. С другой стороны, нелинейные операции, такие как медианная фильтрация, не могут быть реализованы подобным образом.

Для удобства остаток этой главы разделен на несколько частей следующим образом:

- часть А. Обзор операторов низкоуровневой обработки изображений;
- часть В. Выделение и распознавание 2D-объектов;
- часть С. Выделение трехмерных объектов и важность инвариантности;
- часть D. Отслеживание движущихся объектов;
- часть Е. Анализ текстур;
- часть F. От искусственных нейронных сетей к методам глубокого обучения;
- часть G. Заключение.

В целом назначение этой главы состоит в том, чтобы обобщить ключевые понятия и достижения ранних – или «устаревших» – исследований в области компьютерного зрения и напомнить читателям об их значении, чтобы они могли более уверенно освоить новейшие разработки в этой области. Однако необходимость сделать такой выбор означает, что пришлось исключить многие другие важные темы.

1.2. Часть А. ОБЗОР ОПЕРАТОРОВ НИЗКОУРОВНЕВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

1.2.1. Основы обнаружения краев

Обнаружение краев (edge detection) является наиболее важной и широко применяемой операцией обработки изображений. Для этого есть разные важные причины, но в конечном счете описание форм объектов по их краям и внутренним контурам уменьшает объем данных, необходимых для хранения изображения $N \times N$, с $O(N^2)$ до $O(N)$, тем самым значительно повышая эффективность последующего хранения и обработки. Кроме того, хорошо известно, что люди могут очень эффективно распознавать объекты по их контурам (иногда даже лучше, чем по полному изображению): легкость и достоверность распознавания двумерных эскизов и мультфильмов могут служить тому подтверждением.

В 1960-х и 1970-х годах было разработано значительное количество операторов обнаружения краев, многие из которых были в первую очередь интуитивно понятными, а это означает, что их оптимальность была под вопросом. Некоторые операторы применяли 8 или 12 масок-шаблонов для обнаружения краев с разной ориентацией. Как ни странно, прошло достаточно много времени, прежде чем возникло понимание, что, поскольку края являются векторами, для их обнаружения должно быть достаточно двух масок. Однако это не сразу устранило необходимость принятия решения о том, какие коэффициенты маски следует использовать в детекторах краев – даже в случае окрестностей 3×3 , – и мы перейдем к дальнейшему изучению этого вопроса.

Далее мы исходно полагаем, что необходимо использовать 8 масок с углами, отличающимися на 45° . Однако 4 из этих масок отличаются от остальных только знаком, что делает ненужным их отдельное применение. На данный момент аргументы симметрии приводят к следующим маскам для 0° и 45° соответственно:

$$\begin{bmatrix} -A & 0 & A \\ -B & 0 & B \\ -A & 0 & A \end{bmatrix} \quad \begin{bmatrix} 0 & C & D \\ -C & 0 & C \\ -D & -C & 0 \end{bmatrix}. \quad (1.7)$$

Очевидно, что очень важно спроектировать маски так, чтобы они давали правильные ответы в разных направлениях. Чтобы выяснить, как это влияет

на коэффициенты маски, воспользуемся тем фактом, что градиенты интенсивности должны следовать правилам сложения векторов. Если значения интенсивности пикселей в окрестности 3×3 равны

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}, \quad (1.8)$$

представленные выше маски приведут к следующим оценкам градиента в направлениях 0° , 90° и 45° :

$$\begin{aligned} g_0 &= A(c + i - a - g) + B(f - d); \\ g_{90} &= A(a + c - g - i) + B(b - h); \\ g_{45} &= C(b + f - d - h) + D(c - g). \end{aligned} \quad (1.9)$$

Если сложение векторов должно быть допустимым, мы также имеем:

$$g_{45} = (g_0 + g_{90})/\sqrt{2}. \quad (1.10)$$

Приравнивание коэффициентов при a, b, \dots, i приводит к самосогласованной паре условий:

$$\begin{aligned} C &= B/\sqrt{2}; \\ D &= A\sqrt{2}. \end{aligned} \quad (1.11)$$

Далее обратите внимание на дополнительное требование – маски 0° и 45° должны давать одинаковые отклики при $22,5^\circ$. На самом деле за этим утверждением скрываются довольно утомительные алгебраические выкладки (Davies, 1986), которые показывают, что

$$B/A = (13\sqrt{2} - 4)/7 = 2,055. \quad (1.12)$$

Округляя значение этого выражения до 2, мы прямо приходим к маскам оператора Собеля:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad (1.13)$$

применение которого дает карты компонентов g_x, g_y градиента интенсивности. Поскольку края являются векторами, мы можем вычислить локальную величину края g и направление θ , используя стандартные векторные формулы:

$$\begin{aligned} g &= [g_x^2 + g_y^2]^{1/2}; \\ \theta &= \arctan(g_y/g_x). \end{aligned} \quad (1.14)$$

Обратите внимание, что вычисления g и θ для всего изображения не будут свертками, поскольку они включают нелинейные операции.

Итак, в разделах 1.1 и 1.2.1 мы описали различные категории операторов обработки изображений, включая линейные и нелинейные операторы и операторы свертки. Примерами свертков (линейных операций) являются среднее и гауссово сглаживание и оценка компонента краевого градиента. Примерами нелинейных операций являются порог, вычисление краевого градиента и ориентации края. Следует отметить, что коэффициенты маски Собеля были получены в качестве побочного продукта, а не целенаправленно. Фактически они были разработаны для оптимизации точности ориентации краев. Заметим также, что, как мы увидим позже, точность ориентации имеет первостепенное значение, когда информация о краях передается в схемы расположения объектов, такие как преобразование Хафа.

1.2.2. Оператор Кэнни

Детектор краев Кэнни изначально был создан как намного более точная замена основных детекторов краев, таких как детектор Собеля, и вызвал настоящий фурор после публикации в 1986 году (Canny, 1986). Для достижения столь высокой точности по очереди применяется ряд процессов:

1. Изображение сглаживается с помощью двумерного гауссиана, чтобы гарантировать, что поле интенсивности является математически корректной функцией.
2. Изображение дифференцируется с использованием двух одномерных производных функций, таких как функции Собеля, и вычисляется поле величины градиента.
3. Для утончения краев используется немаксимальное подавление вдоль направления нормали локального края. Это происходит в два этапа: (1) нахождение двух нецентральных красных точек, показанных на рис. 1.1, что включает интерполяцию величины градиента между двумя парами пикселей; (2) выполнение квадратичной интерполяции между градиентами интенсивности в трех красных точках для определения положения сигнала края пика с субпиксельной точностью.
4. Выполняется «гистерезисная» пороговая обработка: применение двух порогов t_1 и t_2 ($t_2 > t_1$) к полю градиента интенсивности; результатом является «не край», если $g < t_1$, «край», если $g > t_2$, а иначе это будет «край», только если он находится рядом с «краем». (Обратите внимание, что свойство «край» может распространяться от пикселя к пикселю в соответствии с приведенными выше правилами.)

Как отмечено в пункте 3, для определения местоположения пика амплитуды градиента может использоваться квадратичная интерполяция. Несложные алгебраические выкладки показывают, что для g -значений g_1, g_2, g_3 трех красных точек смещение пика от центральной красной точки равно $(g_3 - g_1) \sec\theta / [2(2g_2 - g_1 - g_3)]$: здесь $\sec\theta$ – это коэффициент, на который θ увеличивает расстояние между крайними красными точками.

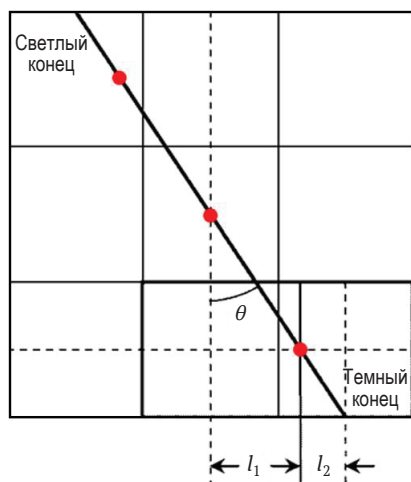


Рис. 1.1 ❖ Использование квадратичной интерполяции для определения точного положения пика амплитуды градиента

1.2.3. Обнаружение сегмента линии

В разделе 1.2.1 мы показали, как при помощи детектора краев всего с двумя масками вычисляется величина и ориентация признака края. Стоит подумать, можно ли использовать аналогичный векторный подход и в других случаях. Действительно, модифицированный векторный подход также можно использовать для обнаружения признаков *сегментов линии*. В этом можно убедиться, рассмотрев следующую пару масок:

$$L_1 = A \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}; \quad L_2 = B \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}. \quad (1.15)$$

Ясно, что можно построить еще две маски такого вида, но они отличаются от двух предыдущих только знаком и ими можно пренебречь. Таким образом, этот набор масок содержит ровно столько, сколько необходимо для векторного вычисления. В самом деле, если мы ищем темные полосы на светлом фоне, 1 может обозначать линию, а -1 может представлять светлый фон. (Нули можно рассматривать как «безразличные» коэффициенты, так как они будут игнорироваться в любой свертке.) Следовательно, L_1 представляет собой полосу 0° , а L_2 – полосу 45° . (Термин «полоса» используется здесь для обозначения сегмента линии значимой ширины.) Применяя тот же метод, что и в разделе 1.2.1, и определяя значения интенсивности пикселей, как в уравнении (1.8), находим:

$$\begin{aligned} l_0 &= A(d + f - b - h); \\ l_{45} &= B(c + g - a - i). \end{aligned} \quad (1.16)$$

Однако в данном случае недостаточно информации для определения отношения A к B , поэтому это должно зависеть от практических аспектов ситуации. Учитывая, что это вычисление выполняется в окрестности 3×3 , неудивительно, что оптимальная ширина полосы для обнаружения с использованием вышеуказанных масок равна 1,0; эксперименты (Davies, 1997) показали, что согласование масок с шириной полосы w (или наоборот) дает оптимальную точность ориентации при $w \approx 1,4$, что имеет место при $B/A \approx 0,86$. Отсюда получается максимальная ошибка ориентации $\sim 0,4^\circ$, что выгодно отличается от $\sim 0,8^\circ$ для оператора Собеля.

Воспользуемся формулами, аналогичными формулам в разделе 1.2.1, для псевдовекторного расчета коэффициента интенсивности линии l и ориентации сегмента линии θ :

$$l = [l_0^2 + l_{45}^2]^{1/2};$$

$$\theta = \frac{1}{2} \arctan(l_{45}/l_0). \quad (1.17)$$

Здесь мы были вынуждены включить коэффициент $1/2$ перед арктангенсом: это потому, что отрезок прямой демонстрирует симметрию вращения на 180° по сравнению с 360° для обычных углов.

Обратите внимание, что это снова тот случай, когда оптимизация направлена на достижение высокой точности ориентации, а не, например, на чувствительность обнаружения.

Здесь стоит отметить два применения обнаружения линейных сегментов. Одним из них является осмотр сыпучих зерен пшеницы для обнаружения мелких темных насекомых, которые напоминают темные полосы: для этого использовались маски 7×7 , разработанные на основе приведенной выше модели (Davies и др., 2003). Другим применением является определение расположения артефактов, таких как телеграфные провода на фоне неба или тросов, поддерживающих киноактеров, которые затем можно целенаправленно удалять.

1.2.4. Оптимизация чувствительности обнаружения

Оптимизация чувствительности обнаружения – задача, которая хорошо известна в радиолокации и очень эффективно применялась для этой цели со времен Второй мировой войны. По сути, эффективное обнаружение летательных аппаратов радиолокационными системами требует оптимизации отношения сигнал–шум (signal to noise ratio, SNR). Конечно, в случае радара обнаружение – это одномерная проблема, тогда как при построении изображений нам необходимо оптимально обнаруживать двумерные объекты на фоне шума. Однако шум изображения не обязательно является гауссовым белым шумом, как обычно предполагается применительно к радару, хотя удобно начать с этого предположения.

В радиолокации сигналы можно рассматривать как положительные пики на фоне шума, который обычно близок к нулю. В этих условиях применима хорошо известная теорема, которая гласит, что оптимальное обнаружение сигнала заданной формы достигается с помощью «согласованного фильтра», который имеет ту же форму характеристики, что и идеализированный входной сигнал. То же самое относится к изображениям, и в этом случае пространственный согласованный фильтр должен иметь ту же форму характеристики, что и идеальная форма искомого двумерного объекта.

Кратко рассмотрим математическую основу этого подхода. Во-первых, мы предполагаем набор пикселей, в которых производится выборка сигналов, что дает значения S_i . Затем мы выражаем желаемый фильтр в виде n -элементного весового шаблона с коэффициентами w_i . Наконец, предполагаем, что уровни шума в каждом пикселе независимы и подчиняются локальным распределениям со стандартными отклонениями N_i .

Очевидно, что суммарный сигнал, полученный от весового шаблона, можно записать в виде:

$$S = \sum_{(i=1)}^n w_i S_i, \quad (1.18)$$

тогда как общий шум, полученный от весового шаблона, будет характеризоваться его дисперсией:

$$N^2 = \sum_{(i=1)}^n w_i^2 N_i^2. \quad (1.19)$$

Следовательно, SNR равно

$$\rho^2 = S^2/N^2 = \left(\sum_{i=1}^n w_i S_i \right)^2 / \sum_{i=1}^n w_i^2 N_i^2. \quad (1.20)$$

Для нахождения оптимального SNR найдем производную

$$\partial \rho^2 / \partial w_i = (1/N^4) [N^2 (2SS_i) - S^2 (2w_i N_i^2)] = (2S/N^4) [N^2 S_i - S(w_i N_i^2)], \quad (1.21)$$

а затем примем $\partial \rho^2 / \partial w_i = 0$ и сразу получим

$$w_i = \frac{S_i}{N_i^2} \times \frac{N^2}{S}, \quad (1.22)$$

что можно записать проще как

$$w_i \propto \frac{S_i}{N_i^2}, \quad (1.23)$$

хотя знак пропорциональности можно заменить равенством без ограничения общности.

Обратите внимание, что если N_i не зависит от i (т. е. уровень шума не меняется на всей площади изображения), то $w_i = S_i$: это доказывает упомянутую выше теорему о том, что *пространственный согласованный* фильтр должен иметь тот же профиль интенсивности, что и двумерный объект, подлежащий обнаружению.

1.2.5. Работа с изменениями интенсивности фона

Помимо очевидной разницы в размерности, есть еще одно важное отличие зрения от радара: у последнего в отсутствие входного сигнала выходной сигнал системы колеблется и в среднем равен нулю. Однако в компьютерном зрении уровень фона обычно будет меняться в зависимости от окружающего освещения, а также в зависимости от входного изображения. По сути, решение этой проблемы заключается в использовании масок с нулевой суммой (или нулевым средним). Поэтому для такой маски, как в уравнении (1.2), мы просто вычитаем среднее значение \bar{c} всех компонентов маски из каждого компонента, чтобы убедиться, что общая маска имеет нулевое среднее значение.

Чтобы убедиться, что использование стратегии нулевого среднего работает, представьте себе применение немодифицированной маски к окрестности изображения, показанной в уравнении (1.3), – допустим, мы получили значение K . Теперь добавим B к интенсивности каждого пикселя в окрестности; это добавит $\sum_n Bc_i = B\sum_n c_i = Bn\bar{c}$ к значению K . Но если мы сделаем $\bar{c} = 0$, то получим исходный вывод маски K .

В целом мы должны отметить, что стратегия нулевого среднего является лишь приближением, так как на изображении будут места, где фон варьируется между высоким и низким уровнями, поэтому невозможно точное устранение нулевого среднего (т. е. B нельзя рассматривать как постоянную над областью маски). Тем не менее если предположить, что изменение фона происходит в масштабе, значительно превышающем масштаб размера маски, эта стратегия должна работать адекватно.

Следует отметить, что аппроксимация с нулевым средним значением уже широко используется, как вы видели на примере масок ребер и сегментов линий в уравнениях (1.7) и (1.15). Этот подход также должен применяться к другим детекторам, таким как детекторы углов и отверстий.

1.2.6. Теория, сочетающая согласованный фильтр и конструкции с нулевым средним

На первый взгляд идея нулевого среднего настолько проста, что может показаться, что она легко интегрируется с формулами согласованного фильтра из раздела 1.2.4. Однако применение нулевого среднего уменьшает количество степеней свободы согласованного фильтра на одну, поэтому необходимо изменить формальное представление согласованного фильтра, чтобы последний продолжал оставаться идеальным детектором. Дабы продолжить,

мы представляем случаи с нулевым средним и согласованным фильтром следующим образом:

$$\begin{aligned}(w_i)_{z-m} &= S_i - \bar{S}; \\ (w_i)_{m-f} &= S_i/N_i^2.\end{aligned}\tag{1.24}$$

Далее мы объединяем их в форму

$$w_i = (S_i - \bar{S})/N_i^2,\tag{1.25}$$

где мы избежали тупика, попробовав гипотетический (т. е. пока неизвестный) тип среднего для S , который мы называем \tilde{S} . (Конечно, если эта гипотеза в конце концов приведет к противоречию, потребуются новый подход.) Применение условия нулевого среднего $\sum_i w_i = 0$ теперь дает следующее:

$$\sum_i w_i = \sum_i S_i/N_i^2 - \sum_i \tilde{S}/N_i^2 = 0;\tag{1.26}$$

$$\therefore \quad \tilde{S} \sum_i (1/N_i^2) = \sum_i S_i/N_i^2;\tag{1.27}$$

$$\therefore \quad \tilde{S} = \sum_i (S_i/N_i^2) / \sum_i (1/N_i^2).\tag{1.28}$$

Из этого мы делаем вывод, что \tilde{S} должно быть взвешенным средним, в частности взвешенным средним по шуму \tilde{S} . С другой стороны, если шум равномерный, \tilde{S} вернется к обычному невзвешенному среднему \bar{S} . Кроме того, если мы не применяем условие нулевого среднего (которого мы можем достичь, установив $\tilde{S} = 0$), уравнение (1.25) сразу возвращается к стандартному условию согласованного фильтра.

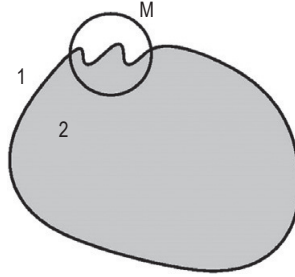
Формула для \tilde{S} может показаться излишне обобщенной, поскольку N_i обычно почти не зависит от i . Однако если бы идеальный профиль был получен путем усреднения профилей реальных объектов, то вдали от его центра дисперсия шума могла бы быть более существенной. Действительно, для больших объектов это было бы явным ограничивающим фактором при таком подходе. Но для относительно небольших объектов и признаков дисперсия шума не должна чрезмерно варьироваться и должны быть достижимы полезные профили согласованного фильтра.

От себя хочу отметить, что основной результат, доказанный в этом разделе (ср. уравнения (1.25) и (1.28)), отнял у меня столько времени и усилий, что я начал было сомневаться в своей способности достичь его. Поэтому я стал называть его «последней теоремой Дэвиса».

1.2.7. Структура маски (дополнительные соображения)

Хотя формальное представление согласованного фильтра и полностью интегрированное к данному моменту условие нулевого среднего могут пока-

заться достаточно общими, чтобы обеспечить однозначную структуру маски, остается ряд аспектов, которые еще предстоит рассмотреть. Например, какого размера должны быть маски? И как их оптимально разместить вокруг каких-либо примечательных объектов или признаков? Чтобы ответить на этот вопрос, мы возьмем следующий пример довольно сложного признака объекта. Здесь область 2 – это обнаруживаемый объект, область 1 – фон, а М – область маски признака.



© IET 1999

В этой модели мы должны рассчитать оптимальные значения весовых коэффициентов маски w_1 и w_2 и площадей областей A_1 и A_2 . Мы можем записать общую мощность сигнала и шума из маски шаблона как:

$$\begin{aligned} S &= w_1 A_1 S_1 + w_2 A_2 S_2; \\ N^2 &= w_1^2 A_1 N_1^2 + w_2^2 A_2 N_2^2. \end{aligned} \quad (1.29)$$

Таким образом, мы получаем отношение мощности сигнал–шум (SNR):

$$f_{i,t+1} = f_{i,t} + \frac{\partial f}{\partial \phi} \frac{\partial \phi}{\partial t}. \quad (1.30)$$

Легко видеть, что если обе области маски увеличить по площади одинаково в η раз, то во столько же раз увеличится и ρ^2 . Следовательно, мы можем оптимизировать маску, регулируя *относительные* значения A_1 и A_2 и оставляя общую площадь A неизменной. Давайте сначала исключим w_2 , используя условие нулевого среднего (которое обычно применяется для предотвращения влияния изменений уровня интенсивности фона на результат):

$$w_1 A_1 + w_2 A_2 = 0. \quad (1.31)$$

Ясно, что мощность SNR больше не зависит от весов маски:

$$\rho^2 = \frac{S^2}{N^2} = \frac{(S_1 - S_2)^2}{N_1^2/A_1 + N_2^2/A_2}. \quad (1.32)$$

Далее, поскольку общая площадь маски A заранее определена, мы имеем:

$$A_2 = A - A_1. \quad (1.33)$$

Подстановка A_2 сразу дает нам простое условие оптимизации:

$$A_1/A_2 = N_1/N_2. \quad (1.34)$$

Принимая $N_1 = N_2$, мы получаем важный результат – *правило равных площадей* (Davies, 1999):

$$A_1 = A_2 = A/2. \quad (1.35)$$

Наконец, когда применяется правило равных площадей, правило нулевого среднего принимает форму:

$$w_1 = -w_2. \quad (1.36)$$

Обратите внимание, что многие случаи, например возникающие, когда передний план и фон имеют разные текстуры, можно смоделировать, полагая $N_1 \neq N_2$. В этом случае правило равной площади не применяется, но мы все еще можем использовать уравнение (1.34).

1.2.8. Обнаружение угла

В разделах 1.2.1 и 1.2.3 мы обнаружили, что только два типа признаков имеют векторную (или псевдовекторную) форму – края и линейные сегменты. Следовательно, в то время как эти признаки могут быть обнаружены с использованием всего лишь двух компонентных масок, ожидается, что все остальные признаки потребуют сопоставления со многими другими шаблонами, чтобы справиться с различными ориентациями. К этой категории относятся и *детекторы углов*, у которых типичные угловые шаблоны 3×3 имеют следующий вид:

$$\begin{bmatrix} -4 & 5 & 5 \\ -4 & 5 & 5 \\ -4 & -4 & -4 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -4 & 5 & -4 \\ -4 & -4 & -4 \end{bmatrix}. \quad (1.37)$$

(Обратите внимание, что эти маски были настроены на форму с нулевым средним значением, дабы устранить эффекты различных условий освещения.)

Чтобы преодолеть очевидные проблемы сопоставления шаблонов, не последней из которых является необходимость использования ограниченного числа цифровых масок для аппроксимации аналоговых вариаций интенсивности, которые сами по себе заметно различаются от экземпляра к экземпляру, было предпринято много усилий по выработке более принципиального подхода. В частности, поскольку края определяются первыми производными поля интенсивности изображения, казалось логичным перейти к производным второго порядка. Одним из первых таких исследований был подход Боде (1978), в котором использовались операторы Лапласа и Гессе:

$$\begin{aligned} \text{Лапласиан} &= I_{xx} + I_{yy}; \\ \text{Гессиан} &= I_{xx}I_{yy} - I_{xy}^2. \end{aligned} \quad (1.38)$$

Они были особенно привлекательны, поскольку определены в терминах детерминанта и следа симметричной матрицы вторых производных и, таким образом, инвариантны относительно вращения.

На самом деле *оператор Лапласа* дает существенные отклики вдоль линий и краев и, следовательно, не особенно подходит для обнаружения углов. С другой стороны, *оператор Боде* (*оператор Гессе*), известный как «DET», не реагирует на линии и края, но дает значимые сигналы вблизи углов и, следовательно, полезен для построения детектора углов, хотя он реагирует одним знаком на одной стороне угла и обратным знаком на другой стороне угла: на самом углу дает нулевой ответ. Кроме того, другие исследователи подвергли критике специфические отклики оператора DET и обнаружили, что им необходим довольно сложный анализ, чтобы определить наличие и точное положение каждого угла (Dreschler, Nagel, 1981; Nagel, 1983).

Тем не менее Китчен и Розенфельд (Kitchen, Rosenfeld, 1982) показали, что они смогли преодолеть эти проблемы, оценив скорость изменения вектора направления градиента вдоль направления касательной горизонтального края и связав его с горизонтальной кривизной k функции интенсивности I . Чтобы получить реалистичное представление о *силе* угла, они умножили k на величину локального градиента интенсивности g :

$$C = \kappa g = \kappa (I_x^2 + I_y^2)^{1/2} = \frac{I_{xx}I_y^2 - 2I_{xx}I_xI_y + I_{yy}I_x^2}{I_x^2 + I_y^2}. \quad (1.39)$$

Наконец, они использовали эвристику не максимального подавления вдоль нормального направления края для дальнейшей локализации угловых положений.

Интересно, что Нагель (Nagel, 1983) и Шах и Джайн (Shah, Jain, 1984) пришли к выводу, что угловые детекторы Китчена и Розенфельда, Дрешлера и Нагеля, а также Зуниги и Харалика (Zuniga, Haralick 1983) по существу эквивалентны. Это не должно вызывать большого удивления, так как, в конце концов, можно было бы ожидать, что различные методы будут отражать одни и те же лежащие в основе физические явления (Davies, 1988) – определение производной второго порядка, которое можно интерпретировать как горизонтальную кривизну, умноженную на градиент интенсивности.

1.2.9. ОПЕРАТОР «ОСОБОЙ ТОЧКИ» ХАРРИСА

На этом этапе Харрис и Стивенс (Harris, Stephens, 1988) разработали совершенно новый оператор, способный обнаруживать признаки угла, основанный не на производных второго порядка, а на производных первого порядка. Как мы увидим ниже, это упростило математическую составляющую, включая избавление от трудностей применения цифровых масок к аналоговым функциям. Фактически новый оператор смог выполнять функцию производной второго порядка, применяя операции первого порядка. Любопытно, ка-

ким образом он извлекает соответствующую информацию о производных второго порядка. Чтобы понять это, нам нужно изучить его довольно простое математическое определение.

Оператор Харриса определяется локальными компонентами градиента интенсивности I_x, I_y в изображении. Определение оператора требует, чтобы область окна была определена и усреднялась $\langle \cdot \rangle$, дабы занять все это окно. Начнем с вычисления следующей матрицы:

$$\Delta = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}. \quad (1.40)$$

Затем мы используем детерминант (det) и след (trace) для оценки углового сигнала:

$$C = \det \Delta / \text{trace } \Delta. \quad (1.41)$$

(Опять же, что касается операторов Боде, значение использования только детерминанта и следа заключается в том, что результирующий сигнал будет инвариантным к угловой ориентации.)

Прежде чем приступить к анализу формы C , заметим, что если бы не проводилось усреднение, $\det \Delta$ был бы тождественно равен нулю: ясно, что только сглаживание, присущее операции усреднения, допускает разброс значений первой производной и тем самым позволяет результату частично зависеть от вторых производных.

Чтобы понять работу детектора в деталях, сначала рассмотрим его отклик для одиночного края (рис. 1.2a). Фактически здесь

$$\det \Delta = 0, \quad (1.42)$$

потому что I_x равен нулю во всей области окна.

Далее рассмотрим ситуацию в окрестностях угла (рис. 1.2b). Здесь:

$$\Delta = \begin{bmatrix} l_2 g^2 \sin^2 \theta & l_2 g^2 \sin \theta \cos \theta \\ l_2 g^2 \sin \theta \cos \theta & l_2 g^2 \cos^2 \theta + l_1 g^2 \end{bmatrix},$$

где l_1, l_2 – длины двух краев, ограничивающих угол, а g – контраст края, предполагаемый постоянным для всего окна. Теперь мы находим (Davies, 2005):

$$\det \Delta = l_1 l_2 g^4 \sin^2 \theta, \quad (1.44)$$

а также

$$\text{trace } \Delta = (l_1 + l_2) g^2; \quad (1.45)$$

$$\therefore C = \frac{l_1 l_2}{l_1 + l_2} g^2 \sin^2 \theta. \quad (1.46)$$

Это можно интерпретировать как произведение (1) коэффициента добротности λ , который зависит от длин кромок в пределах окна, (2) коэффициента контрастности g^2 и (3) коэффициента формы $\sin^2\theta$, который зависит от «резкости» края θ . Ясно, что C равно нулю при $\theta = 0$ и $\theta = \pi$ и максимально при $\theta = \pi/2$ – все эти результаты интуитивно верны и уместны.

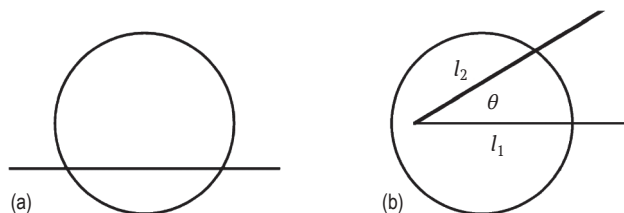


Рис. 1.2 ❖ Геометрическая иллюстрация расчета отклика линии и угла в круглом окне: (а) прямой край, (б) угол в общем виде. © IET 2005

Из этой формулы можно определить многие свойства оператора, в том числе тот факт, что пиковый сигнал возникает не в самом углу, а в центре окна, используемого для вычисления углового сигнала, хотя смещение уменьшается по мере того, как снижается острота угла.

1.3. Часть В. Локализация и распознавание ДВУХМЕРНЫХ ОБЪЕКТОВ

1.3.1. Подход к анализу формы на основе центроидного профиля

Двухмерные объекты обычно характеризуются формой их границ. В этом разделе мы рассмотрим, чего можно достичь, отслеживая границы объекта и анализируя полученные профили формы. Среди наиболее распространенных типов профилей, используемых для этой цели, выделяется *центроидный профиль*, в котором граница объекта наносится на карту с использованием полярных координат (r, θ) , принимая центроид C границы за начало координат.

В случае круга радиуса R центроидный профиль представляет собой прямую линию на расстоянии R выше оси θ . На рис. 1.3 представлено пояснение, а также показаны два примера разбитых круглых объектов. В случае (а) окружность лишь слегка искривлена, и поэтому ее центроид C остается практически неизменным; следовательно, большая часть центроидного графика остается на расстоянии R выше оси θ . Однако в случае (б) даже та

часть границы, которая не нарушена и не искажена, находится далеко не на постоянном расстоянии от оси θ : это означает, что объект невозможно узнать по его профилю, хотя в случае (а) нетрудно распознать его как слегка поврежденный круг. На самом деле мы уделяем столько внимания этим случаям в основном из-за того факта, что в случае (b) центростид смещается так сильно, что даже неизменная часть формы не может быть немедленно распознана. Конечно, можно было бы попытаться исправить ситуацию, переместив центростид обратно в положение, соответствующее кругу, но это довольно сложная задача: во всяком случае, если исходная фигура не является кругом, много вычислений будет потрачено впустую до того, как станет понятна истинная природа проблемы.

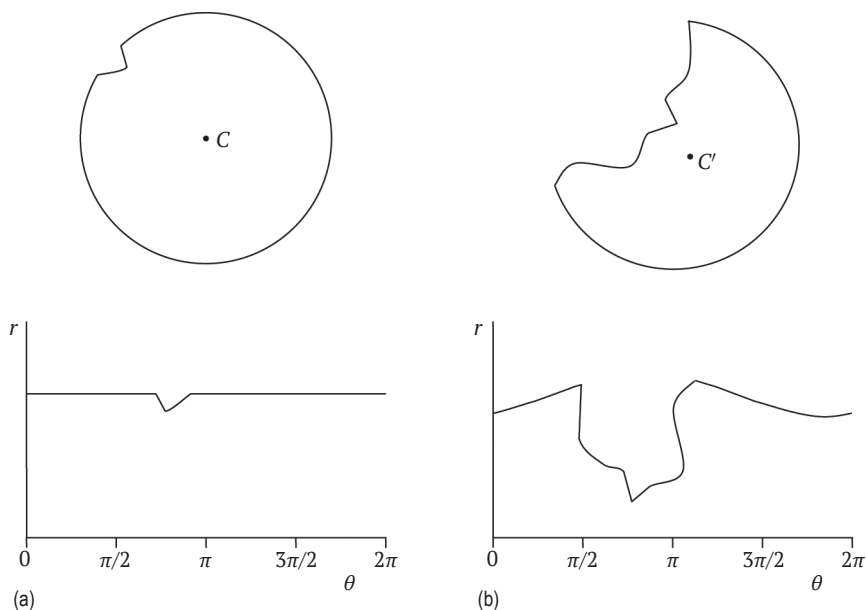


Рис. 1.3 ❖ Проблемы с дескриптором центроидального профиля: (а) представлен круглый объект с небольшим дефектом на его границе; под ним изображен соответствующий центроидный профиль; (b) представлен тот же объект, но на этот раз с грубым дефектом: поскольку центростид смещен в сторону C' , весь профиль центроида сильно искажен

В целом мы можем заключить, что подход с центроидным профилем ненадежен и не рекомендуется. На самом деле это не означает, что его совсем не следует использовать на практике. Например, на конвейере для сыра или печенья любой предмет, который не распознается сразу по постоянному R -профилю, должен быть немедленно удален с конвейера; затем можно исследовать оставшиеся объекты более тщательно, чтобы убедиться, что их значения R приемлемы и демонстрируют надлежащую степень постоянства.

РОБАСТНОСТЬ И ЕЕ ЗНАЧЕНИЕ

Не случайно здесь возникла идея *робастности*¹. Она лежит в основе большей части дискуссий о ценности и эффективности алгоритмов, имеющих прямое отношение к компьютерному зрению. Основная проблема заключается в изменчивости объектов или любых иных сущностей, присущей компьютерным изображениям. Эта изменчивость может возникать по совершенно разным причинам: шум, различная форма объектов (даже одного и того же типа), различия в размере или расположении, трещины или дефекты, разное расположение камер и разные режимы просмотра. Кроме того, один объект может быть частично затенен другим или только частично находиться в определенном изображении (что дает эффекты, не отличающиеся от механического повреждения объекта).

Хотя хорошо известно, что шум влияет на точность измерения, можно подумать, что он с меньшей вероятностью повлияет на робастность. Однако нам необходимо отличать «обычный» тип шума, который мы можем описать как *гауссов шум*, от пикового или импульсного шума. Последние обычно описываются как выделяющиеся точки или «выбросы» в распределении шума. (Напомню, что мы уже видели, как медианный фильтр значительно лучше справляется с выбросами, чем средний фильтр.) Предметом *робастной статистики* является изучение темы нормальных значений и выбросов, а также то, как лучше всего справляться с различными типами шума. Исследования в этой области лежат в основе оптимизации точности измерения и достоверности интерпретации при наличии выбросов и грубых нарушений внешнего вида объекта.

Далее следует отметить, что существуют другие типы графических представлений границ, которые можно использовать вместо центроидного профиля. Один из них представляет собой график (s, ψ) , а другой – производный профиль (s, κ) . Здесь ψ – угол ориентации границы, а $\kappa(s)$, равный $d\psi/ds$, – локальная функция кривизны. Важно отметить, что эти представления не основаны на положении центроида, следовательно, его положение не нужно вычислять или даже оценивать. Несмотря на это преимущество, все такие представления граничных профилей имеют еще одну существенную проблему: если какая-либо часть границы закрыта, искажена или нарушена, сравнение формы объекта с шаблонами известной формы становится весьма затруднительным из-за разной длины границ.

Несмотря на эти проблемы, в подходящих ситуациях метод центроидного профиля имеет определенные преимущества, поскольку он способствует простоте измерения радиусов окружностей, легкости идентификации квадратов и других форм с выступающими углами и простому измерению ориентации, особенно для формы с выступающими углами.

Теперь осталось найти метод, который мог бы заменить метод центроидного профиля в тех случаях, когда могут возникать грубые искажения или *окклюзии* (загораживания одних объектов другими). В поисках такого метода мы переходим к следующему разделу, который знакомит с подходом преобразования Хафа.

¹ Под робастностью в статистике понимают нечувствительность к различным отклонениям и неоднородностям в выборке, связанным с теми или иными, в общем случае неизвестными, причинами. © academic.ru.

1.3.2. Схемы обнаружения объектов на основе преобразования Хафа

В разделе 1.3.1 мы рассмотрели, как круглые объекты могут быть идентифицированы по их границам с использованием подхода центроидного профиля к анализу формы. Этот подход оказался ненадежным из-за его неспособности справиться с грубыми искажениями формы и окклюзиями. В этом разделе мы покажем, что *преобразование Хафа* (Hough Transform) обеспечивает простой, но изящный способ решения данной проблемы. Используемый метод состоит в том, чтобы взять каждую краевую точку на изображении, переместить ее внутрь на расстояние R вдоль локальной нормали к краю и сохранить эту точку в отдельном изображении, называемом *пространством параметров*: R принимается за ожидаемый радиус кругов, которые должны быть локализованы. Результатом этого будет скопление точек (часто называемых «голосами») вокруг местоположений центров кругов. Фактически для получения точных оценок местоположений центров необходимо только найти значимые пики в пространстве параметров.

Этот процесс проиллюстрирован на рис. 1.4, из которого видно, что метод игнорирует некруглые части границы и идентифицирует только настоящие центры окружностей. Таким образом, подход фокусируется на данных, которые соответствуют выбранной модели, и не обращает внимания на нерелевантные данные, которые в противном случае приводят к значительному снижению робастности. Разумеется, данный метод зависит от точности оценки направлений нормалей к краям. К счастью, оператор Собеля способен оценивать ориентацию края с точностью до 1° , и его легко применять. Как показано на рис. 1.5, результаты могут быть весьма впечатляющими.

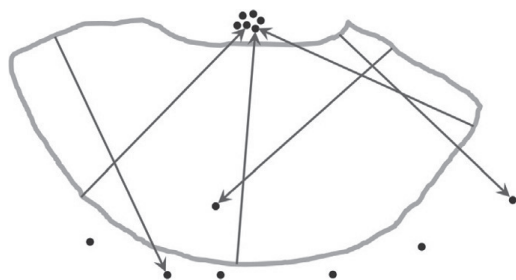


Рис. 1.4 ❖ Робастность преобразования Хафа при нахождении центра круглого объекта. Круглая часть границы дает центральные точки-кандидаты, которые фокусируются на истинном центре, тогда как неправильная ломаная граница дает центральные точки-кандидаты в случайных положениях. В данном случае граница примерно совпадает с границей сломанного печенья, показанного на рис. 1.5

Недостаток описанного выше подхода заключается в том, что ему требуется заранее известное значение R . Общее решение этой проблемы состоит в использовании трехмерного пространства параметров, в котором третье

измерение представляет возможные значения R , и последующем поиске наиболее значимых пиков в этом пространстве. Однако более простое решение включает в себя накопление результатов для диапазона вероятных значений R в одном и том же двумерном пространстве параметров – процедура, которая приводит к существенной экономии памяти и вычислений (Davies, 1988). На рис. 1.6 показан результат применения этой стратегии, которая работает как с положительными, так и с отрицательными значениями R . С другой стороны, в плоскости с одним параметром информация о радиальном расстоянии теряется из-за накопления всех голосов. Следовательно, потребуются дополнительная итерация процедуры для определения радиуса, соответствующего местоположению каждого пика.

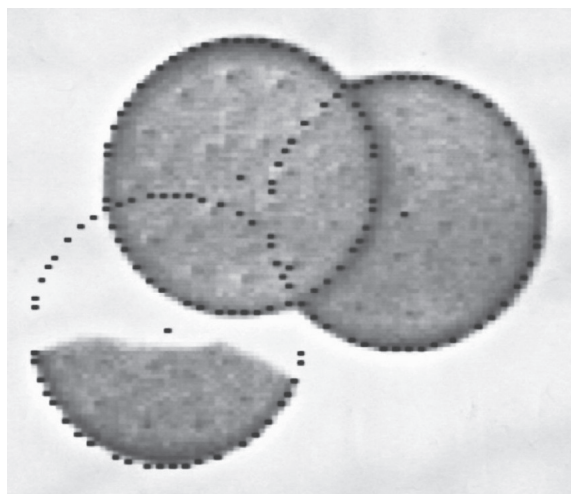


Рис. 1.5 ❖ Набор сломанных и перекрывающихся печений, демонстрирующий надежность метода определения центра. На точность метода указывают черные точки, каждая из которых находится в пределах $1/2$ пикселя радиального расстояния от центра. © IFC 1984

Подход с преобразованием Хафа также можно использовать для обнаружения эллипса: два простых метода для этого случая представлены на рис. 1.7. Оба они воплощают непрямой подход, в котором используются *пары* краевых точек. В то время как *метод бисекции диаметра* требует значительно меньше вычислений, чем *метод хорд и касательных*, он более подвержен ложным обнаружениям, например когда два эллипса лежат рядом друг с другом на изображении.

Чтобы доказать правильность метода хорд и касательных, укажем на применимость этого метода для окружностей, а далее *свойство проективности* гарантирует, что он также сработает для эллипсов, потому что при ортогональной проекции прямые линии проецируются в прямые, средние точки в средние, касательные в касательные, а окружности в эллипсы; кроме того, всегда можно найти такую точку обзора, что окружность можно спроецировать на заданный эллипс.

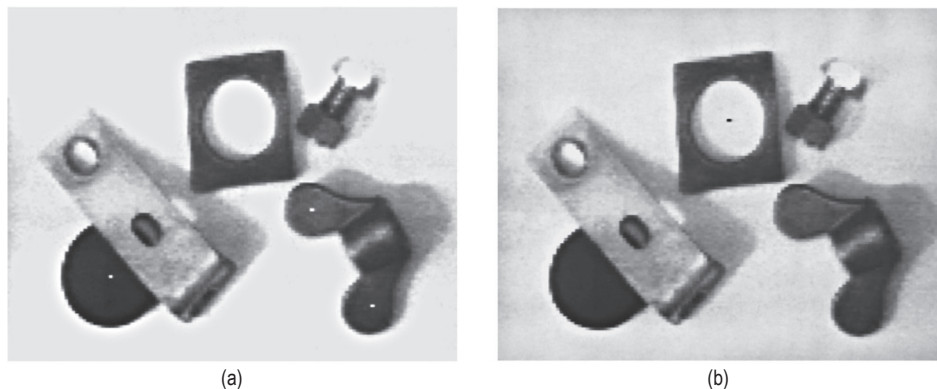


Рис. 1.6 ❖ Одновременное обнаружение объектов с разными радиусами: (а) обнаружение крышки объектива и барашковой гайки, когда предполагается, что радиусы находятся в диапазоне 4–17 пикселей; (б) обнаружение отверстий на том же изображении, когда предполагается, что радиусы попадают в диапазон от –26 до –9 пикселей (используются отрицательные радиусы, поскольку отверстия считаются объектами отрицательного контраста): ясно, что на *этом* изображении мог быть применен меньший диапазон отрицательных радиусов

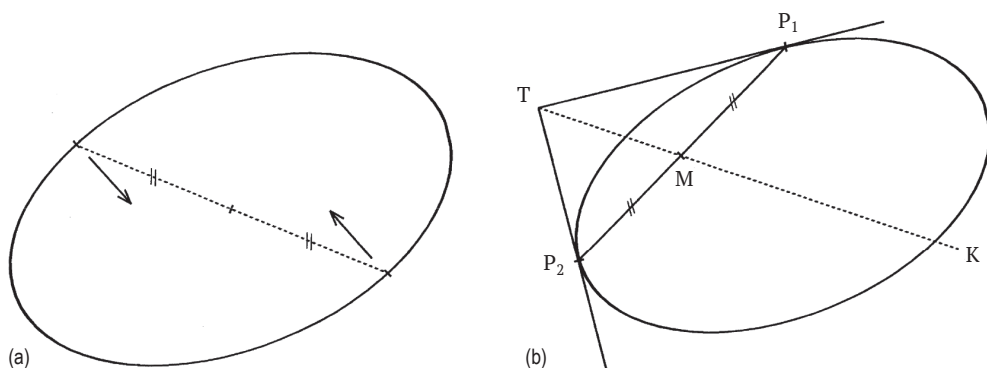


Рис. 1.7 ❖ Геометрическое представление двух методов обнаружения эллипсов: (а) в методе бисекции диаметра находят пару точек, для которых ориентации ребер антипараллельны. Середины таких пар накапливаются, и полученные пики принимаются за центры эллипсов; (б) в методе хорд и касательных касательные в точках P_1 и P_2 пересекаются в точке T , а середина отрезка P_1P_2 находится в точке M . Центр эллипса S лежит на полученной линии TM

Теперь мы переходим к так называемому *обобщенному преобразованию Хафа* (generalized Hough transform, GHT), которое использует более прямую процедуру обнаружения эллипса, чем два других метода, описанных выше.

Чтобы понять, как обобщается стандартный метод Хафа для локализации объектов произвольной формы, нам сначала нужно выбрать точку локализации L в шаблоне идеализированной формы. Затем нам нужно сделать так, чтобы вместо перемещения от краевой точки на фиксированное расстояние

R непосредственно вдоль локальной нормали от края до центра, как в случае с окружностями, мы перемещались на соответствующее *переменное* расстояние R в *переменном* направлении φ так, чтобы прийти к L ; R и φ теперь являются функциями направления нормали к локальному краю θ (рис. 1.8). В этих условиях голоса будут иметь пик в заранее выбранной точке локализации объекта L . Функции $R(\theta)$ и $\varphi(\theta)$ могут быть представлены аналитически в компьютерном алгоритме, а для совершенно произвольных форм они могут быть сохранены в виде интерполяционных таблиц. В любом случае схема основана на очень простом принципе, но при обобщении метода Хафа возникает важное усложнение, потому что мы переходим от изотропной формы (круг) к анизотропной форме, которая может иметь совершенно произвольную ориентацию.

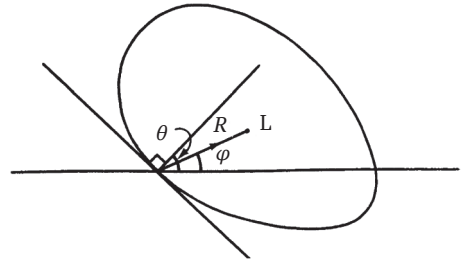


Рис. 1.8 ❖ Вычисление обобщенного преобразования Хафа

Это означает добавление дополнительного измерения в пространство параметров (Ballard, 1981). Затем каждая точка края вносит свой вклад в набор голосов в каждой плоскости ориентации в пространстве параметров. Наконец, все пространство параметров просматривается в поисках пиков – наивысших точек, указывающих как на расположение объектов, так и на их ориентацию. Интересно, что ГНТ может обнаруживать эллипсы, используя одну плоскость в пространстве параметров, за счет применения *функции точечной экстраполяции* (point spread function, PSF) к каждой краевой точке, которая учитывает все возможные ориентации эллипса: обратите внимание, что PSF применяется на некотором расстоянии от краевой точки, чтобы центр PSF мог пройти через центр эллипса (рис. 1.9). Ограниченный объем главы не позволяет представить здесь детали вычислений (например, см. Davies, 2017, глава 11).

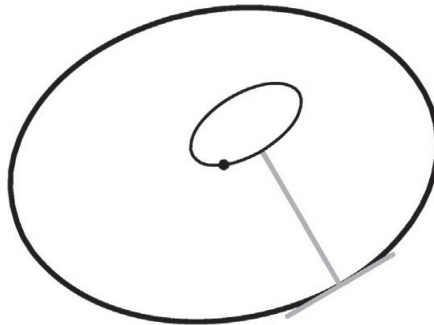


Рис. 1.9 ❖ Использование формы PSF, учитывающей все возможные ориентации эллипса. PSF позиционируется серыми вспомогательными линиями так, чтобы она проходила через центр эллипса (черная точка)

1.3.3. Применение преобразования Хафа для обнаружения линий

Преобразование Хафа (НТ) также может применяться для обнаружения линий. Ранее было отмечено, что лучше избегать обычного уравнения с коэффициентом наклона и точкой пересечения вида $y = mx + c$, потому что для почти вертикальных линий требуются почти бесконечные значения m и c . Вместо этого использовалась «нормальная» (θ, ρ) форма прямой линии (рис. 1.10):

$$\rho = x \cos \theta + y \sin \theta. \quad (1.47)$$

Для применения метода в этой форме множество прямых, проходящих через каждую точку P_i , представляют в виде множества синусоид в пространстве (θ, ρ) : например, для точки $P_1(x_1, y_1)$ синусоида имеет уравнение:

$$\rho = x_1 \cos \theta + y_1 \sin \theta. \quad (1.48)$$

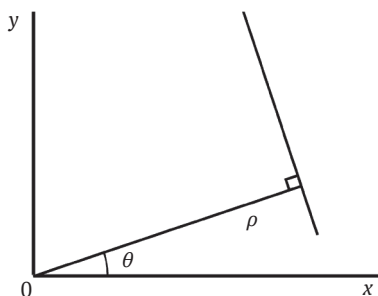


Рис. 1.10 ❖ Нормальная параметризация прямой линии в пространстве (θ, ρ)

После накопления голосов в пространстве (θ, ρ) пики указывают на наличие линий в исходном изображении.

Была проделана большая работа (см., например, Dudani, Luk, 1978) для ограничения погрешностей определения местоположения линии, возникающих по разным причинам: шум, дискретизация, эффекты фрагментации линии, эффекты небольшой кривизны линии, сложность оценки точных положений пиков в пространстве параметров. Кроме того, важна проблема локализации продольной линии. Для последнего из этих процессов Дудани и Лук (Dudani, Luk, 1978) разработали метод «ху-группировки», который предусматривал проведение анализа связности для каждой линии. Затем сегменты линии подлежали объединению, если они разделены промежутками менее ~5 пикселей. Наконец, сегменты короче определенной минимальной длины (также обычно ~5 пикселей) игнорировались как слишком незначительные, чтобы облегчить интерпретацию изображения.

В целом мы видим, что все описанные выше формы НТ значительно выигрывают благодаря наличию механизма *накопления доказательств* (accumulating evidence) с использованием схемы голосования. Этот механизм является источ-

ником высокой робастности метода. Вычислительные процессы, используемые НТ, можно описать скорее как индуктивные, а не дедуктивные, поскольку наличие пиков приводит к *гипотезам* о присутствии объектов, которые в принципе должны быть подтверждены другими доказательствами, тогда как *дедукция* привела бы к немедленному доказательству присутствия объектов.

1.3.4. Использование RANSAC для обнаружения линий

RANSAC – это альтернативная схема поиска на основе моделей, которую часто можно использовать вместо НТ. Дело в том, что она очень эффективно работает при обнаружении линий, поэтому заслуживает отдельного внимания. Стратегию поиска можно рассматривать как схему голосования, но она используется иначе, чем в НТ. Она выдвигает последовательность гипотез о целевых объектах и определяет поддержку каждой из них, подсчитывая, сколько точек данных согласуется с ними в разумных (например, $\pm 3\sigma$) пределах (см. рис. 1.11). Как и следовало ожидать, для любого потенциального искомого объекта на каждом этапе сохраняются только гипотезы с максимальной поддержкой.

Давайте разберем, как RANSAC используется для обнаружения линий. Как и в случае с НТ, мы начинаем с применения детектора краев и определения местоположения всех краевых точек на изображении. Как мы увидим, RANSAC лучше всего работает с ограниченным количеством точек, поэтому полезно найти краевые точки, которые являются локальными максимумами градиента интенсивности изображения. Далее, все, что необходимо, чтобы сформулировать гипотезу прямой линии, – это взять любую пару граничных точек. Для каждой гипотезы мы проходим по списку N краевых точек, определяя, сколько точек M поддерживают гипотезу. Затем мы берем другие гипотезы (другие пары краевых точек) и на каждом этапе оставляем только ту, которая дает максимальную поддержку M_{\max} . Этот процесс показан в листинге 1.1.

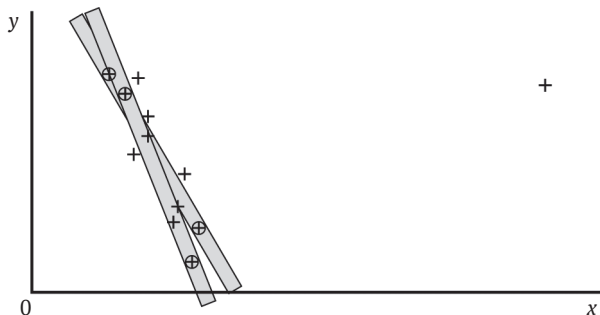


Рис. 1.11 ❖ Метод RANSAC. Здесь знаки + указывают точки данных, по которым нужно попытаться подогнать линии, а также показаны два экземпляра пар точек данных (обозначенных знаками ⊕), через которые проведены гипотетические линии. Каждая предполагаемая линия имеет область допуска $\pm t$, в пределах которой ищется поддержка максимального количества точек данных. Линия с наибольшей поддержкой считается наиболее подходящей

Листинг 1.1 ❖ Базовый алгоритм RANSAC для поиска линии с наибольшей поддержкой. Этот алгоритм возвращает только одну линию; точнее, он возвращает модель линии, которая имеет наибольшую поддержку. Линии с меньшей поддержкой в итоге игнорируются

```

Mmax=0;
для всех пар краевых точек {
    найти уравнение линии, определяемое двумя точками i, j;
    M = 0;
    для всех N точек в списке
        если (точка k находится в пределах порогового расстояния d от линии) M++;
    если (M > Mmax) {
        Mmax = M;
        imax = i;
        jmax = j;
        // это гипотеза, имеющая максимальную поддержку на данный момент
    }
}
/* если Mmax > 0, (x[imax], y[imax]) и (x[jmax], y[jmax]) будут координатами точек,
определяющих линию с наибольшей поддержкой */

```

Алгоритм в листинге псевдокода 1.1 соответствует поиску центра самого высокого пика в пространстве параметров, как и в случае НТ. Чтобы найти все линии на изображении, наиболее очевидной стратегией является следующая: найти первую линию, затем удалить все точки, поддерживающие ее; потом найти следующую линию и устранить все точки, поддерживающие ее; повторять, пока все точки не будут исключены из списка. Процесс может быть записан более компактно в таком виде:

```

повторить {
    найти линию;
    удалить поддерживающие точки;
}
пока не закончатся точки данных;

```

Как сказано выше, RANSAC предполагает довольно значительную вычислительную нагрузку, составляющую $O(N^3)$, по сравнению с $O(N)$ для алгоритма НТ. Следовательно, при использовании RANSAC лучше каким-то образом уменьшить N . Это объясняет, почему полезно сосредоточиться на локальных максимумах, а не использовать полный список краевых точек. Однако в качестве альтернативы можно использовать повторную случайную выборку из полного списка до тех пор, пока не будет проверено достаточное количество гипотез, чтобы быть уверенным в том, что обнаружены все значимые линии. К слову, эти идеи отражают первоначальное значение аббревиатуры RANSAC, которая расшифровывается как RANdom SAMpling Consensus – консенсус с произвольными данными, в том смысле, что любая гипотеза должна формировать консенсус с доступными подтверждающими данными (Fischler, Bolles, 1981). Степень уверенности в том, что все значимые линии обнаружены, можно вычислить как обратную величину риска того, что значимая линия будет пропущена из-за пропуска репрезентативной пары точек, лежащих на линии.

Теперь мы можем рассмотреть результаты, полученные путем применения RANSAC к частному случаю поиска прямых линий. В описанном тесте в качестве гипотез использовались пары точек, а все краевые точки представляли собой локальные максимумы градиента интенсивности. Случай, показанный на рис. 1.12, соответствует обнаружению деревянного бруска в форме икосаэдра. Обратите внимание, что одна линия справа на рис. 1.12a была пропущена, потому что пришлось установить нижний предел уровня поддержки для каждой линии: это было необходимо, потому что ниже этого уровня поддержки количество случайных коллинеарностей резко возросло даже для относительно небольшого числа краевых точек, показанных на рис. 1.12b, что приводит к резкому увеличению числа ложноположительных линий. В целом этот пример показывает, что RANSAC является очень важным претендентом на определение местоположения прямых линий в цифровых изображениях. Здесь не обсуждается тот факт, что RANSAC полезен для получения надежной подгонки ко многим другим типам форм как в 2D, так и в 3D.

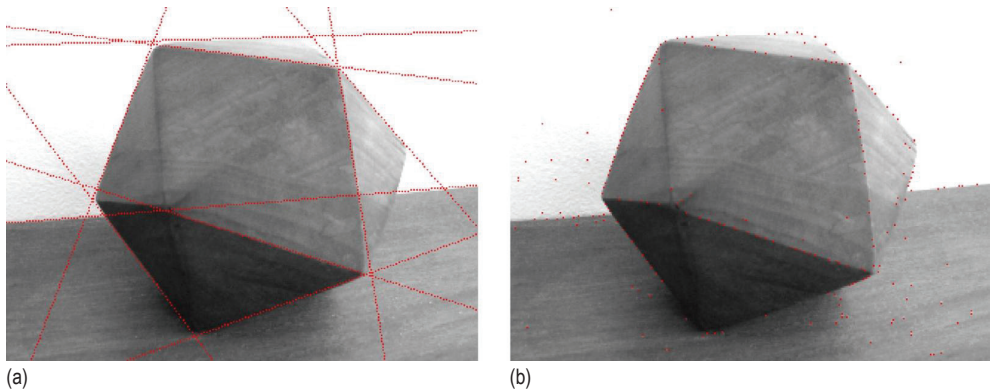


Рис. 1.12 ❖ Обнаружение прямых линий с использованием метода RANSAC: (a) исходное изображение в оттенках серого с прямыми краевыми линиями, обнаруженными с использованием метода RANSAC: (b) краевые точки, переданные в RANSAC для получения (a): это были локальные максимумы градиентов изображения. В (a) пропущены три ребра икосаэдра. Это потому, что они представляют собой края с низким контрастом и низким градиентом интенсивности. Фактически RANSAC также упустил четвертый край из-за наличия нижнего предела уровня поддержки (см. текст выше)

Наконец, следует упомянуть, что RANSAC менее, чем HT, подвержен влиянию алиасинга (ступенчатого искажения) вдоль прямых линий. Это связано с тем, что пики HT, как правило, фрагментируются из-за алиасинга, поэтому наилучшие гипотезы трудно получить без агрессивного сглаживания изображения. Причина, по которой RANSAC выигрывает в этом контексте, заключается в том, что он полагается не на точность отдельных гипотез, а скорее на их количество: стратегия исходит из того, что достаточное количество гипотез можно легко генерировать и столь же легко отбрасывать.

1.3.5. Теоретико-графовый подход к определению положения объекта

В этом разделе мы рассмотрим часто встречаемую ситуацию со значительными ограничениями – объекты появляются на горизонтальном рабочем столе или конвейере на известном расстоянии от камеры. Также предполагается, что (а) объекты плоские или могут располагаться только в ограниченном количестве позиций в трех измерениях, (b) объекты рассматриваются вертикально сверху и что (с) искажения перспективы малы. В подобных ситуациях объекты в принципе могут быть идентифицированы и локализованы по очень небольшому количеству точечных признаков. Поскольку считается, что такие признаки не имеют собственной структуры, будет невозможно однозначно определить положение объекта по одному признаку, хотя положительная идентификация и определение положения были бы возможны с использованием двух признаков, если бы они были различимы и если бы было известно их расстояние друг от друга. Если говорить о действительно неразличимых точечных признаках, невозможно устранить неоднозначность для всех объектов, не обладающих 180-градусной симметрией вращения. Следовательно, как правило, для идентификации и определения объектов на известном расстоянии требуются как минимум три точечных признака. Очевидно, что шум и другие артефакты, такие как окклюзии, ухудшают ситуацию. Фактически при сопоставлении шаблона точек идеализированного объекта с точками, присутствующими на реальном изображении, мы обнаруживаем, что:

- 1) из-за нескольких экземпляров выбранного типа объекта на изображении могут присутствовать очень много характерных точек;
- 2) из-за шума или помех от посторонних объектов и структур на заднем плане могут присутствовать лишние точки;
- 3) некоторые точки, которые должны присутствовать, отсутствуют из-за шума или окклюзии, или из-за дефектов искомого объекта.

Эти проблемы означают, что мы должны пытаться сопоставить подмножество точек в идеализированном шаблоне с различными подмножествами точек на изображении. Если считать, что наборы точек составляют графы с точечными признаками в качестве узлов, задача превращается в математическую проблему изоморфизма подграфов, т. е. нахождения того, какие подграфы в графе изображения изоморфны подграфам идеализированного шаблонного графа. (Изоморфность означает наличие одинаковой базовой формы и структуры.) Ясно, что схема сопоставления точечных признаков будет наиболее успешной, если она находит наиболее вероятную интерпретацию путем поиска решений, обладающих наибольшей внутренней согласованностью, т. е. с наибольшим числом совпадений точек на объекте.

К сожалению, представленная выше схема все еще слишком проста для многих применений, поскольку она недостаточно устойчива к искажениям. В частности, могут возникать оптические (например, перспективные) искажения, сами объекты могут быть деформированы или, опираясь частично

на другие объекты, могут принять нестандартное положение, в силу чего расстояния между признаками могут быть не совсем такими, как ожидалось. Эти факторы означают, что должен существовать некоторый допуск в отношении расстояний между парами признаков. Ясно, что искажения создают дополнительную нагрузку на технику сопоставления точек и делают еще более необходимым поиск решений с максимально возможной внутренней устойчивостью. Поэтому при обнаружении и идентификации объектов следует учитывать как можно больше признаков. Для этого предназначен метод *максимальной клики* (maximal clique).

Для начала на исходном изображении идентифицируется как можно больше признаков: обычно они нумеруются в порядке появления на телевизионном растре. Затем числа должны быть сопоставлены с буквами, соответствующими признакам идеализированного объекта. Систематическим способом достижения этого является построение *графа соответствия* (match graph), или, как его еще называют, *графа ассоциации* (association graph), в котором узлы представляют соотнесения признаков, а дуги, соединяющие узлы, представляют попарную совместимость между соотнесениями. Чтобы найти наилучшее соответствие, необходимо найти области графа соответствия, где перекрестные связи максимальны. Для этого в графе соответствий ищутся клики. *Клика* – это *полный подграф*, т. е. такой, у которого все пары узлов соединены дугами. Однако предыдущие аргументы указывают на то, что если одна клика полностью включена в другую клику, вполне вероятно, что более крупная клика представляет собой лучшее совпадение – и действительно, максимальные клики можно рассматривать как ведущие к наиболее надежным совпадениям между наблюдаемым изображением и моделью объекта.

На рис. 1.13а показана ситуация для общего четырехугольника, его график соответствия показан на рис. 1.13б. В этом случае есть 16 возможных соотнесений признаков, 12 допустимых совместимостей и 7 максимальных клик. Если происходит перекрытие признака, оно (взятое само по себе) уменьшит количество возможных отнесений признаков, а также количество допустимых совместимостей: кроме того, количество максимальных клик и размер наибольшей максимальной клики будут уменьшены. С другой стороны, шум или беспорядок могут добавить ошибочные признаки. Если последние находятся на произвольном расстоянии от существующих признаков, то количество возможных отнесений признаков будет увеличено, но совместимости в графе соответствий больше не будет, так они привнесут лишь тривиальную дополнительную сложность. Однако если дополнительные признаки появляются на *допустимом* расстоянии от существующих признаков, это добавит дополнительную совместимость в граф соответствия и сделает его более нагруженным для анализа. В случае, показанном на рис. 1.14, возникают оба типа осложнений – окклюзия и дополнительный признак: теперь имеется 8 парных отнесений и 6 максимальных клик, что в целом несколько меньше, чем в исходном случае на рис. 1.13. Однако важным фактором является то, что наибольшая максимальная клика по-прежнему указывает на наиболее вероятную интерпретацию изображения, поэтому метод по своей природе очень надежен.

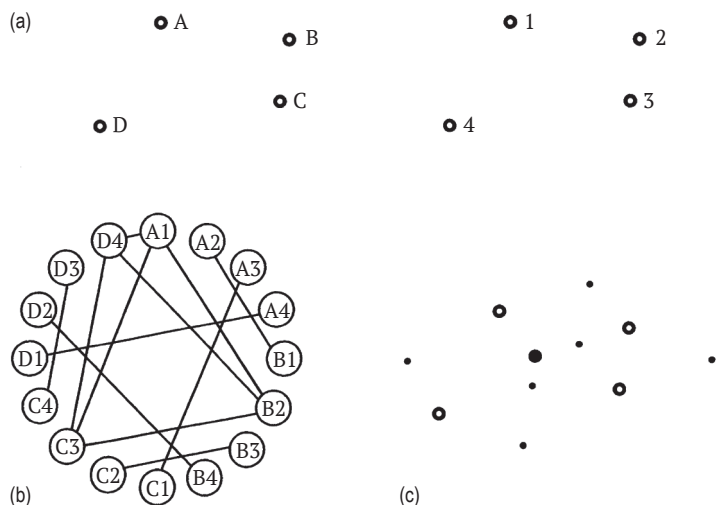


Рис. 1.13 ❖ Задача сопоставления для общего четырехугольника: (а) базовая маркировка модели (слева) и изображения (справа); (б) граф сопоставления; (с) размещение голосов в пространстве параметров: маленькие кружки обозначают положение отверстий, точки обозначают отдельные голоса, а большая точка показывает положение основного пика. © АВК 1988

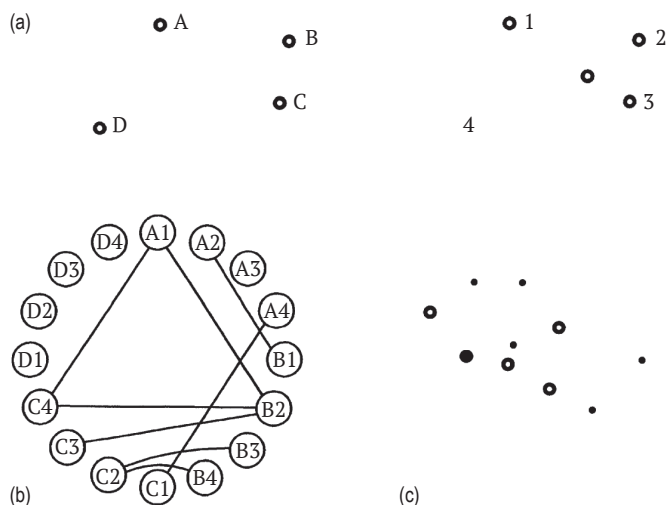


Рис. 1.14 ❖ Сопоставление при перекрытии одного объекта и добавлении другого: (а) базовая маркировка модели (слева) и изображения (справа); (б) графическое соответствие; (с) размещение голосов в пространстве параметров (обозначения, как на рис. 1.13)

На рис. 1.15а показана пара печений, которые должны быть обнаружены по их «докерным» отверстиям, – эта стратегия выгодна, поскольку она позволяет очень точно определить местонахождение продукта до детального осмотра. Отверстия, обнаруженные простой процедурой сопоставления

с шаблоном, показаны на рис. 1.15а: используемый шаблон довольно мал, и в результате процедура работает достаточно быстро, но не может найти все отверстия; кроме того, она может давать ложные срабатывания. Следовательно, для анализа данных о местоположении отверстия необходимо использовать «интеллектуальный» алгоритм. Анализ данных в приведенном выше примере дает две нетривиальные максимальные клики, каждая из которых правильно соответствует одному из двух печений на изображении.

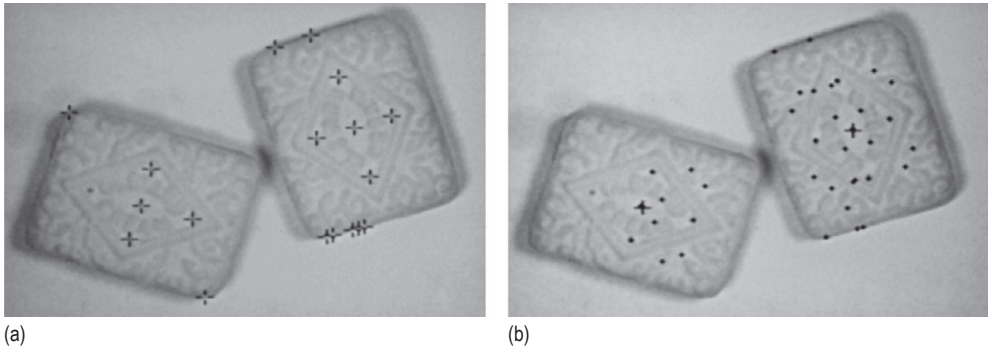


Рис. 1.15 ❖ Поиск печений и выяснение их расположения: (а) два печенья с крестиками, указывающими на результат применения простой процедуры обнаружения отверстий; (б) два печенья, надежно обнаруженных ГНТ по данным отверстий из (а): изолированные маленькие крестики указывают позиции одиночных голосов. © АВК 1988

1.3.6. Использование обобщенного преобразования Хафа для экономии вычислений

В предыдущих примерах проверка того, какие подграфы являются максимальными кликами, является простой задачей. К сожалению, время выполнения оптимального алгоритма максимальной клики ограничено не многочленом от M (для графа соответствия, содержащего максимальные клики до M узлов), а гораздо более быстро меняющейся функцией. В частности, известно, что задача поиска максимальных клик является NP -полной и время ее выполнения растет экспоненциально. Таким образом, каким бы ни было время выполнения для значений M примерно до 6, оно обычно будет в 100 раз больше для значений M примерно до 10 и еще в 100 раз больше для M больше 14.

Далее мы покажем, как в качестве альтернативы подходу максимальной клики можно использовать обобщенное преобразование Хафа (ГНТ). Чтобы применить ГНТ, мы сначала перечисляем все признаки, а затем накапливаем голоса в пространстве параметров в каждой возможной позиции точки локализации L , соответствующей каждой паре признаков (рис. 1.16). Чтобы проделать это, необходимо просто использовать межэлементное расстояние

в качестве параметра поиска в R -таблице ГНТ. Для неразличимых точечных признаков это означает, что должны быть две записи для положения L для каждого значения расстояния между признаками. Процедура иллюстрируется примером обобщенного четырехугольника на рис. 1.13: это приводит к 7 пикам в пространстве параметров, веса которых равны 6, 1, 1, 1, 1, 1, 1 (см. рис. 1.13с). Аналогичная ситуация возникает и для рис. 1.14. Внимательное изучение рис. 1.13 и 1.14 показывает, что каждый пик в пространстве параметров соответствует максимальной клике в графе соответствия. Действительно, между ними существует взаимно однозначное отношение, поэтому все правильные совместимости вносят вклад как в большую максимальную клику, так и в большой пик в пространстве параметров. Эта стратегия по-прежнему применима, даже когда возникают окклюзии или присутствуют дополнительные признаки (см. рис. 1.14).

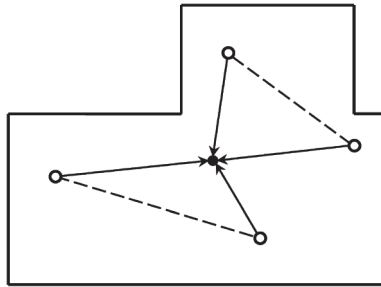


Рис. 1.16 ❖ Метод определения местоположения L по парам позиций признаков: каждая пара точек признаков дает две возможные позиции голосования в пространстве параметров, когда объекты не имеют симметрии. При наличии симметрии определенные пары признаков могут дать до 4 позиций для голосования: это подтверждается при внимательном изучении рис. 1.15b

Наконец, снова рассмотрим пример на рис. 1.15а, на этот раз получив решение с помощью ГНТ. На рис. 1.15b показаны положения центров объектов-кандидатов, найденные с помощью ГНТ. Маленькие изолированные крестики указывают позиции одиночных голосов, а те, что очень близки к двум большим крестикам, приводят к пикам голосования весов 10 и 6 в этих соответствующих позициях. Следовательно, местоположение объекта является точным и надежным, как и требуется (Davies, 1988b).

Теперь сравним вычислительные требования подходов максимальной клики и ГНТ к определению расположения объекта. Для простоты представьте себе изображение, которое содержит только один полностью видимый пример объекта, обладающего n признаками, и что мы пытаемся распознать его, ища все возможные попарные совместимости.

Для объекта, обладающего n признаками, граф соответствий содержит n^2 узлов (т. е. возможных назначений), и существует $n^2 C_2 = n^2(n^2-1)/2$ возможных попарных совместимостей, которые необходимо проверить при построении графа. Объем вычислений на этом этапе анализа составляет $O(n^4)$. К этому следует добавить вычислительную стоимость нахождения макси-

мальных клик. Поскольку задача является NP-полной, вычислительная нагрузка растет со скоростью, близкой к экспоненциальной по n^2 .

Теперь рассмотрим затраты на GHT при поиске объектов с помощью попарной совместимости. Как мы видели, общая высота всех пиков в пространстве параметров равна количеству попарных совместимостей в графе соответствий. Следовательно, вычислительная нагрузка имеет тот же порядок, $O(n^4)$. Далее возникает задача локализации всех пиков в пространстве параметров. Для изображения $N \times N$ необходимо посетить только N^2 точек в пространстве параметров, а вычислительная нагрузка составляет $O(N^2)$, хотя постоянное ведение записи о максимальном местоположении во время голосования может значительно уменьшить ее (Davies, 1988b).

1.3.7. Подходы на основе частей

В то время как подходы к определению местоположения объектов, описанные выше, как правило, рассчитаны на поиск соответствия объектов вполне определенным геометрическим моделям, существуют и совершенно другие подходы, которые заключаются в использовании таких методов, как *деформируемые модели* (deformable models). Эти подходы учитывают различия во внешнем виде, возникающие в результате изменений освещения, точки обзора и таких свойств, как форма и цвет. В качестве наиболее важных примеров можно назвать поиск лиц или пешеходов в дорожных сценах. К методам этой категории относятся *жесткие шаблоны* (rigid templates; Dalal, Triggs, 2005), *пакет признаков* (bag-of-features; Zhang et al., 2007), деформируемые шаблоны (например, Cootes, Taylor, 2001) и модели на основе частей (part-based models; например, Amit, Trounev, 2007; Leibe et al., 2008). Модели деформируемых частей обучаются с использованием наборов частей, расположенных в деформируемых конфигурациях. Этот подход вышел на первый план в 2010 г., когда было показано (Felzenszwalb et al., 2010), что это приводит к эффективным, точным и современным результатам на сложных наборах данных.

Модели деформируемых частей (deformable parts models, DPM) основаны на идее, что объекты можно рассматривать как наборы частей. Таким образом, для обнаружения таких объектов, как лица, необходимо только определить местонахождение частей и изучить их взаимосвязь. Это может быть выполнено путем определения частей и их ограничивающих рамок, а затем внесения предложений по объединению их в более крупные ограничивающие рамки, представляющие объекты. По сути, после того как были найдены ограничивающие объекты рамки, эти области защищаются от дальнейшего анализа неадекватным подавлением. На практике это означает, что каждой потенциальной ограничивающей рамке присваивается оценка, сохраняется наивысшая оценка и пропускаются те рамки, которые перекрывают уже существующую ограничивающую рамку, на критический процент, например 50 %. Этот очень успешный подход достиг самых современных результатов в тестах PASCAL VOC 2006, 2007 и 2008 (Everingham et al., 2006; 2007; 2008) и «зарекомендовал себя как стандарт де-факто для обнаружения общих объ-

ектов» (Mathias et al., 2014). Матиас с коллегами очень тщательно протестировали подход DPM и показали, что он может обеспечить максимальную производительность при распознавании лиц.

Интересно, что подход DPM позволял обнаруживать явно трехмерные объекты, но без необходимости непосредственного учета их трехмерной геометрии, что достигается путем достаточно разнообразного обучения на соответствующих типах объектов. Еще одна полезная возможность заключается в том, что этот подход также может быть очень эффективным для обнаружения сочлененных объектов.

Подход DPM очень важен, поскольку он лег в основу подходов глубокого обучения с еще более высокими рейтингами производительности (например, Bai et al., 2016) (подробнее об этом будет сказано в части F про методы глубокого обучения).

1.4. Часть C. РАСПОЛОЖЕНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ И ВАЖНОСТЬ НЕИЗМЕННОСТИ

1.4.1. Введение в трехмерное зрение

В предыдущих частях этой главы обычно предполагалось, что объекты по существу плоские и рассматриваются таким образом, что существует только три степени свободы, а именно две, связанные с положением, и еще одна, связанная с ориентацией. Хотя этого подхода достаточно для решения многих полезных задач компьютерного зрения, он не подходит для интерпретации большинства сцен на открытом воздухе или в помещении, или даже для помощи в довольно простых задачах по роботизированной сборке и осмотру. За последние несколько десятилетий было разработано и подкреплено экспериментами значительное количество теорий, раскрывающих механизмы детального понимания сцен, состоящих из реальных трехмерных объектов.

В целом речь идет о попытках интерпретировать сцены, в которых объекты могут появляться в совершенно произвольных положениях и ориентациях, что соответствует шести степеням свободы. Интерпретация таких сцен и вывод параметров перемещения и ориентации произвольных наборов объектов требуют значительного объема вычислений – отчасти из-за естественной неоднозначности при выводе трехмерной информации из двухмерных изображений. Однако для работы с трехмерным зрением разработано множество подходов, и для успешной интерпретации трехмерных сцен часто требуются тонкие их комбинации.

Прежде чем двигаться дальше, приведем уравнение изображения *общей точки* (X, Y, Z) в сцене при так называемой *перспективной проекции*; оно дает точку изображения:

$$(x, y) = (f X/Z, f Y/Z), \quad (1.49)$$

где f – фокусное расстояние используемого объектива.

Теперь рассмотрим принцип работы зрительной системы человека – *бинокулярное зрение*. Система камер, которая используется для этой цели, изображена на рис. 1.17. В этом геометрическом представлении общая точка обозначена на двух изображениях как (x_1, y_1) и (x_2, y_2) . Как правило, две оптические системы не обязательно должны иметь параллельные оптические оси и будут иметь ненулевой угол схождения (*вергенции*). Однако часто для простоты используется случай нулевой вергенции; мы тоже им воспользуемся. Обратите внимание, что два набора координат, соответствующих общей точке (X, Y, Z) в сцене, будут различаться, поскольку базовая линия b между оптическими осями вызывает относительное смещение, или «несоответствие», точек на двух изображениях.

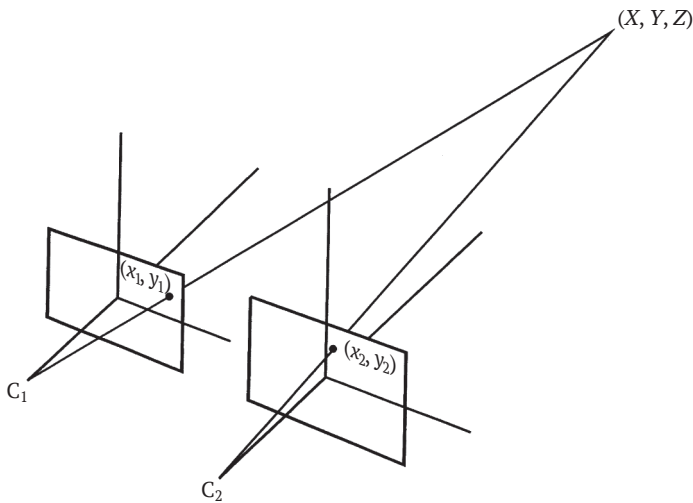


Рис. 1.17 ❖ Стереозображение с использованием двух объективов.

Оси оптических систем параллельны, т.е. между оптическими осями нет схождения

Далее, при подходящем выборе оси Z на серединном перпендикуляре к базовой линии b , получаем два уравнения:

$$x_1 = (X + b/2)f/Z; \quad (1.50)$$

$$x_2 = (X - b/2)f/Z. \quad (1.51)$$

Вычисление расхождения $D = x_1 - x_2$ позволяет сразу получить глубину Z :

$$Z = bf/(x_1 - x_2). \quad (1.52)$$

Хотя это кажется идеальным способом работы с трехмерным зрением, существует фундаментальная проблема – наличие подтверждения того, что обе точки в стереопаре действительно соответствуют одной и той же точке исходной сцены. Заметим также, что для получения высокой точности определения глубины требуется большая базовая линия b : к сожалению, с увеличением b соответствие между изображениями уменьшается, поэтому найти

совпадающие точки становится труднее: это связано с тем, что два изображения становятся все более разными и трудно сопоставимыми.

Стандартным способом решения упомянутой выше проблемы стереосоответствия является метод *эпиполярной линии*, показанный на рис. 1.18. Чтобы понять его суть, представьте, что мы определили местонахождение характерной точки на первом изображении и что мы отмечаем все возможные точки в поле объекта, которые могли ее породить. Мы получим линию точек на разной глубине сцены, и при просмотре во второй плоскости изображения можно построить геометрическое место точек в этой плоскости. Это геометрическое место (на альтернативном изображении) представляет собой эпиполярную линию, соответствующую исходной точке изображения. Если мы теперь будем искать сходную отличительную точку на втором изображении только вдоль эпиполярной линии, шансы найти правильное совпадение значительно возрастут. Этот метод не только сокращает количество вычислений, необходимых для поиска соответствующих точек, но также значительно снижает количество ложных срабатываний. В простой схеме на рис. 1.17 все эпиполярные линии параллельны оси x , хотя это применимо только для случая нулевой сходимости. Отметим, что проблема соответствия значительно усложняется фактом наличия в сцене точек, которые порождают точки на одном изображении, но не на другом: это может возникнуть из-за окклюзии или грубого искажения одной из точек. Таким образом, необходимо искать непротиворечивые множества решений в виде непрерывных поверхностей объектов в сцене.

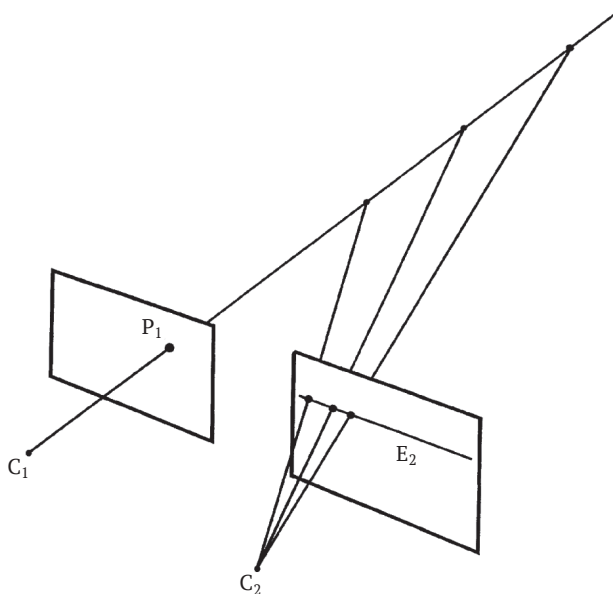


Рис. 1.18 ❖ Схематическая иллюстрация метода эпиполярных линий. Точка P_1 в одной плоскости изображения может возникнуть из любой линии точек сцены и может появиться в альтернативной плоскости изображения в любой точке на так называемой эпиполярной линии E_2

Трудности, вызванные проблемой соответствия, привели к появлению ряда альтернативных подходов. Одним из наиболее известных является «форма из затенения» – выяснение того, как меняется ориентация поверхности, путем анализа наблюдаемой яркости поверхности. Хотя этот подход хорошо себя проявил, он основан на предположении об отражательной способности и текстуре поверхностей и о том, как они меняются в зависимости от ориентации (а) поверхности и (b) источника освещения. Он также требует применения сложных итерационных алгоритмов, но мы не имеем возможности раскрыть здесь эту тему. Точно так же за рамки данной книги выходит метод «фотометрического стереозрения», который подразумевает поочередное освещение сцен отдельными источниками света и анализ полученных изображений.

Подход «форма из текстуры» позволяет анализировать детали ориентации поверхности путем изучения относительных областей текстурных элементов. Однако это специализированный метод, который применяется не очень часто. Подход «структурированного освещения» основан на проецировании световых полос или других схем световых пятен или сеток на поле объекта. Этот подход чрезвычайно широко используется для проверки и сборки объектов на заводских линиях, хотя нельзя сказать, что он широко используется в других областях, таких как наблюдение. Поэтому он тоже является специализированным, а не общим методом измерения формы трехмерных объектов.

Следует отметить, что все эти подходы, кроме последнего, ведут к созданию карт ориентации поверхности, а не к измерению глубины объекта как таковой, поэтому расчет глубины и формы поверхности должен быть выведен из необработанных измерений ориентации.

В целом описанные выше методы используют различные средства для оценки глубины во всех местах сцены и, следовательно, способны отображать трехмерные поверхности с достаточной степенью детализации. Однако они не дают никакого понимания того, что представляют собой эти поверхности. В некоторых ситуациях может быть ясно, что определенные плоские поверхности являются частями фона, например пол и стены комнаты, но в общем случае отдельные объекты не могут быть идентифицированы с абсолютной уверенностью. Действительно, объекты имеют тенденцию сливаться друг с другом и с фоном, поэтому необходимы специальные методы для сегментации трехмерной «карты пространства» и, наконец, распознавания объектов, т. е. предоставления подробной информации об их положении и ориентации. Очевидно, что получение карты глубины трехмерного объекта приводит к его распознаванию не ближе, чем карта границ, такая как центроидальный профиль, к распознаванию двумерного объекта: для выполнения распознавания необходимо разработать специальные средства. К сожалению, в случае трехмерных объектов эта задача значительно сложнее по сравнению с двумерными. Например, хотя преобразование Хафа, в принципе, можно применить в обоих случаях, в трехмерном варианте оно намного сложнее и требует больше вычислений, чем в двумерном, поскольку количество свободных параметров обычно увеличивается с 3 до 6 для статической формы без неизвестных параметров формы – имеется три степени свободы для

перемещения и три для вращения. Заметим также, что вычислительная сложность обычно зависит не от количества степеней свободы, а от показателя степени, равного этому количеству.

Следует упомянуть еще один момент: за последние несколько лет были разработаны датчики, которые обеспечивают выходные данные в формате RGB-D (цвет и глубина): они предоставляют информацию о глубине на основе оптического «времени пролета» – времени, за которое импульс света проходит расстояние до поверхности объекта и обратно. Лидары широко распространены, но стоят дорого и лучше работают на больших расстояниях, в то время как матричные времяпролетные камеры лучше работают на коротких расстояниях и обычно используют генерируемые лазером световые импульсы с разницей в несколько наносекунд. Эти продвинутые современные устройства помогают решить проблему стереосоответствия. Тем не менее они не устраняют проблему интерпретации трехмерных поверхностей, обладающих большим количеством степеней свободы, с которыми приходится бороться алгоритмам идентификации объектов. Напротив, они лишь подчеркивают значимость этой наиболее существенной из оставшихся задач.

Учитывая, что основным источником проблем является вычислительная сложность, было бы естественно исследовать каждую карту глубины на наличие существенных признаков и соответствующим образом интерпретировать сцены: как только у нас будут описания объектов, основанные на относительно небольшом количестве существенных признаков, а не на объемных описаниях поверхностей, появится надежда на достижение быстрой и надежной идентификации. Мы рассмотрим эту возможность далее в следующем разделе.

1.4.2. Неоднозначность положения при перспективной проекции

В этом разделе мы определяем *слабую* и *полную перспективу* и стремимся понять *проблему n -точечной перспективы* (perspective n -point, PnP) – задачу нахождения положения объектов по n признакам при различных формах перспективы.

Полноперспективная проекция (full perspective projection, FPP) – это основная и наиболее общая форма проецирования объекта в изображение, приводящая, например, к тому, что параллельные линии больше не кажутся параллельными, а большинство фигур кажутся искаженными – круги даже выглядят как эллипсы. *Слабая перспективная проекция* (weak perspective projection, WPP) – это форма перспективной проекции, которая используется для удаленных объектов, для которых $\Delta Z \ll Z$. Ее можно рассматривать как «масштабированную ортогональную проекцию» – ортогональная проекция представляет собой тип проекции, который возник бы, если бы объекты проецировались ортогонально параллельными лучами на плоскость изображения, в то время как масштабирование учитывает уменьшение видимого размера объекта.

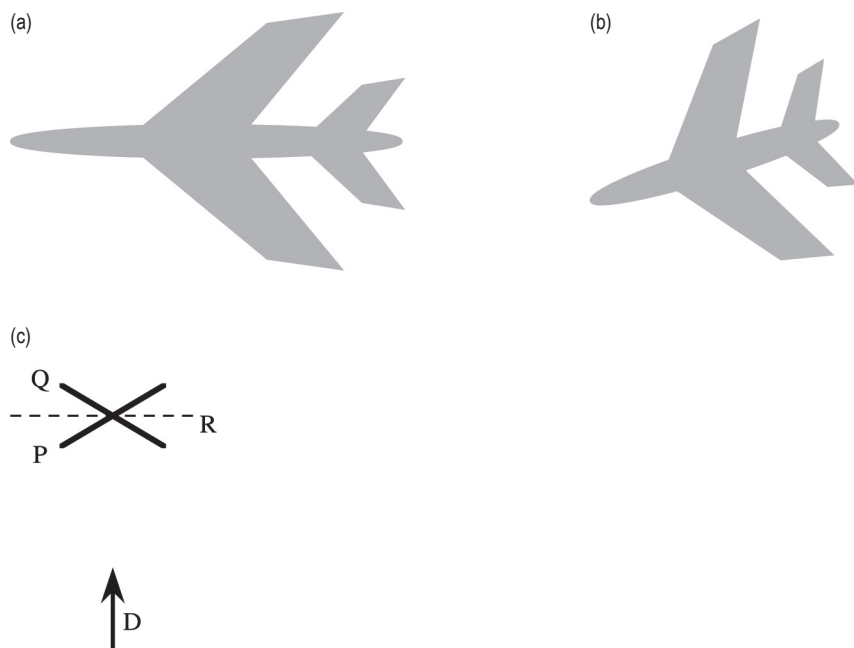


Рис. 1.19 ❖ Перспективная инверсия самолета. Здесь силуэт самолета (а) вырисовывается на фоне неба и выглядит так же, как на (b). На (с) показаны две плоскости Р и Q, в которых мог бы лежать самолет, относительно направления наблюдения D: R – плоскость отражения, объединяющая плоскости Р и Q

Поскольку WPP не искажает формы объектов (например, задняя часть проволочного куба имеет тот же размер и форму, что и его передняя часть), его проще использовать для моделирования процесса визуализации. Однако эта форма проекции настолько проста, что может привести к неоднозначности при просмотре плоских объектов. Это показано на рис. 1.19, на котором видно, что двухмерному виду удаленного самолета могут соответствовать две ориентации, поскольку из одного вида определяется только косинус наклона плоскости α . Интересно, что когда α отличен от нуля, FPP вносит дополнительное искажение в форму самолета, и тогда можно определить истинную ориентацию.

В табл. 1.1 показана вся полнота ситуации, когда плоские объекты обнаруживаются по одному или нескольким их признакам. Эта таблица отражает общую проблему PnP, упомянутую выше. Из таблицы видно, что в компланарном случае (в котором все n признаков объекта компланарны) WPP никогда не дает однозначной интерпретации, тогда как FPP дает, но только когда n больше 3. Причина, по которой n должно быть больше 3, для того чтобы FPP давала однозначный результат, заключается в том, что она включает в себя так много параметров, что трех признаков недостаточно для разрешения ситуации; однако когда присутствуют 4 или более признаков, может быть разрешена полная ситуация и устранена неоднозначность. Но почему WPP не может добиться этого? Причина в том, что в WPP расположение любых

дополнительных признаков (выше 3) может быть выведено из первых трех, поэтому они не могут дать никакой дополнительной информации: следовательно, WPP не может привести к устранению неоднозначности.

Таблица 1.1. Неоднозначности при оценке положения объекта по точечным признакам. В этой таблице сведены количества решений, которые будут получены при оценке положения жесткого объекта по точечным признакам, расположенным на одном изображении. Предполагается, что n точечных признаков обнаружены и идентифицированы правильно и в правильном порядке. Столбцы WPP и FPP означают слабую перспективную проекцию и полную перспективную проекцию соответственно. Верхняя половина таблицы применяется, когда все n точек лежат в одной плоскости; нижняя половина таблицы применяется, когда n точек не компланарны. Заметим, что при $n \leq 3$ результаты строго применимы только в компланарном случае. Однако две верхние строки в нижней половине таблицы сохранены для удобства сравнения

Расположение точек	n	WPP	FPP
Компланарное	≤ 2	∞	∞
	3	2	1
	4	2	1
	5	2	1
	≥ 6	2	1
Некомпланарное	≤ 2	∞	∞
	3	2	4
	4	1	2
	5	1	2
	≥ 6	1	1

Обратите внимание, что когда n равно 1 или 2, существует по крайней мере одна вращательная степень свободы, поэтому существует бесконечное количество решений. На данный момент мы рассмотрели все возможности в верхней половине таблицы, где задействованы компланарные объекты. Далее мы обратимся к некомпланарному случаю, рассмотренному в нижней половине таблицы. Случаи, когда $n \leq 3$, уже рассматривались в верхней половине таблицы, поэтому случай некомпланарности включает только случаи, где $n > 3$.

Давайте теперь рассмотрим, что происходит, когда 4 признака просматриваются в WPP. Взяв по очереди два признака, мы можем создать две плоскости. Когда просматриваются три признака на каждой из этих плоскостей, они генерируют два решения с разными значениями α , и существует только одно согласованное решение. Этот вывод завершает наше толкование записей WPP в табл. 1.2 и, в частности, некомпланарных случаев. Ситуация хорошо иллюстрируется на рис. 1.20 (а–с). Обратите внимание, однако, на спасительную ситуацию, показанную на рис. 1.20d, где видно, как объект, обладающий особой симметрией, может быть подвержен остаточной неоднозначности.

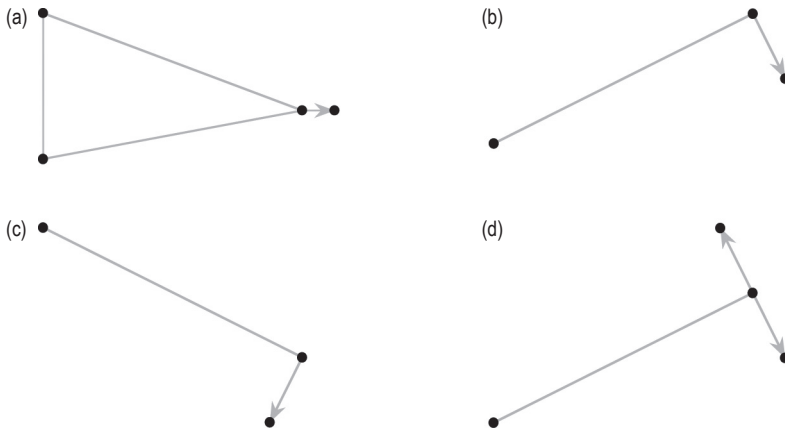


Рис. 1.20 ❖ Определение положения для 4 точек, просматриваемых при слабой перспективной проекции. На (а) показан объект, содержащий четыре некомпланарные точки, наблюдаемый при проекции со слабой перспективой, (б) показывает вид сбоку этого объекта. Если бы первые три точки (соединенные серыми линиями без стрелок) рассматривались отдельно, инверсия перспективы привела бы ко второй интерпретации (с). Однако четвертый пункт дает дополнительную информацию о положении, которая допускает только одну общую интерпретацию. Это не относится к объекту, содержащему дополнительную симметрию, как в (d), поскольку его отражение будет идентично исходному виду (не показано)

Чтобы полностью понять ситуацию, в которой при FPP просматривается более 3 некомпланарных точек, нам необходимо учитывать тот факт, что существует 11 параметров калибровки камеры (см. раздел 1.4.10), которые необходимо определить из 12 линейных однородных уравнений. Это означает, что для вычисления всех 11 параметров в общем случае потребуется не менее 6 некомпланарных точек (включая 2×6 координат изображения). Таким образом, хотя FPP усложняет ситуацию, она также предоставляет больше информации, с помощью которой в конечном итоге можно разрешить неоднозначность.

Наконец, следует подчеркнуть, что приведенные выше рассуждения предполагают, что все соответствия между признаками объекта и изображения известны, т. е. что n точечных признаков обнаруживаются и идентифицируются правильно и в правильном порядке. Если это не так, то количество возможных решений может существенно возрасти, учитывая количество возможных перестановок весьма небольшого числа точек. В качестве одного из способов ограничения этой проблемы можно отметить, что копланарные точки, рассматриваемые в слабой или полной перспективной проекции, всегда появляются в одном и том же циклическом порядке. Если учитывать возможные искажения объекта, то проверка этого утверждения нетривиальна. Впрочем, если выпуклый многоугольник можно провести через точки, циклический порядок вокруг его границы не изменится при проецировании, потому что *плоская* выпуклость является инвариантом проецирования. Однако при некомпланарном расположении точек объекта их воспринимаемая

на изображении структура может перестраиваться почти случайным образом: это означает, что для некомпланарных точек придется учитывать значительно большее количество перестановок точек, чем для компланарных. Еще одно соображение заключается в том, что характерные точки, используемые для распознавания объектов, не должны быть коллинеарными или иметь какой-либо особый шаблон и должны быть описаны как находящиеся в общем положении: в противном случае существует риск того, что некоторые неоднозначности не будут устранены, как указано в табл. 1.2 (в основном потому, что при попытке определить параметры калибровки камеры возникают необратимые уравнения).

1.4.3. Инварианты как помощь в трехмерном распознавании

Инварианты важны для распознавания объектов как в 2D, так и в 3D. Основная идея инварианта состоит в том, чтобы найти некоторый параметр или параметры, которые не изменяются между различными экземплярами или положениями объекта, и использовать их для облегчения идентификации объекта. Как мы увидим, перспектива значительно усложняет задачу в общем трехмерном случае.

Сначала рассмотрим плоский объект, наблюдаемый строго сверху камерой, оптическая ось которой перпендикулярна плоскости, на которой лежит объект. Рассмотрим два точечных признака объекта, таких как углы или небольшие отверстия. Если мы измерим расстояние между признаками в изображении, оно будет действовать как инвариант в том смысле, что:

- 1) имеет величину, не зависящую от параметров перемещения и ориентации объекта;
- 2) будет неизменным для разных объектов одного типа;
- 3) в общем случае будет отличаться от соответствующих параметров других объектов, лежащих на этой же плоскости.

Таким образом, измерение расстояния обеспечивает определенное качество поиска или индексирования, которое в идеале однозначно идентифицирует объект, хотя потребуется дальнейший анализ, чтобы полностью определить его местоположение и установить его ориентацию. Следовательно, межэлементное расстояние имеет все требования 2D-инварианта. Конечно, здесь мы игнорируем неточность измерения из-за неадекватного пространственного разрешения, шума, искажений объектива и т. д.; кроме того, игнорируются эффекты частичной окклюзии или поломки. Очевидно, что есть предел возможностей одной инвариантной меры. В частности, она не справляется с вариациями масштаба объекта. Приближение камеры к плоскости объекта и связанная с этим перефокусировка полностью меняют ситуацию, и все значения инварианта расстояния, находящиеся в таблице индексации объектов, должны быть изменены, а старые значения проигнорированы. Однако эта проблема легко преодолима. Все, что нам нужно сделать, – это взять *отношения* расстояний. Для этого требуется идентифицировать как

минимум 3 точечных признака на изображении и измерить 2 расстояния между признаками. Если мы назовем эти два расстояния d_1 и d_2 , то отношение d_1/d_2 будет действовать как инвариант, не зависящий от масштаба, т. е. мы сможем идентифицировать объекты с помощью одной операции индексации независимо от их двумерного перемещения, ориентации, видимого размера или масштаба.

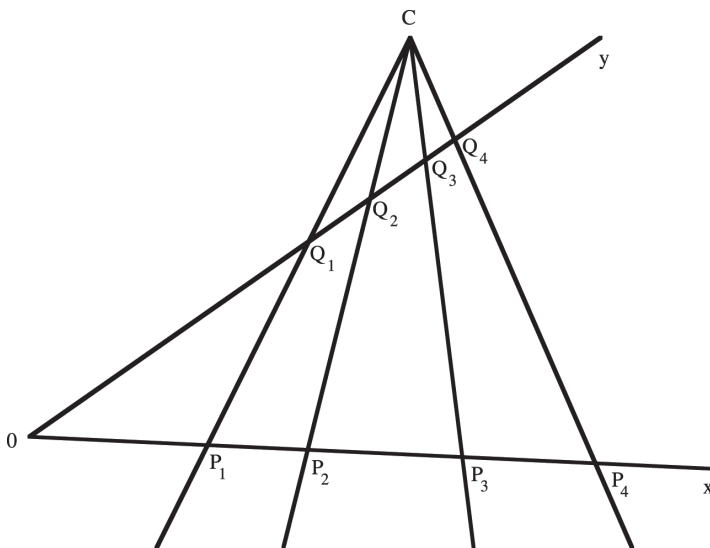


Рис. 1.21 ❖ Перспективное преобразование (perspective transformation) четырех коллинеарных точек. На этом рисунке показаны четыре коллинеарные точки (P_1, P_2, P_3, P_4) и их преобразование (Q_1, Q_2, Q_3, Q_4), аналогичное тому, которое производится системой формирования изображения с оптическим центром С

В целом основная идея использования инвариантов состоит в том, чтобы получить математические измерения конфигураций признаков объекта, которые не зависят от используемой точки обзора или системы координат: действительно, ввиду очевидных сложностей, связанных с перспективной проекцией, независимость точки зрения является решающим фактором в 3D-распознавании объектов и требует использования инвариантов перспективы.

1.4.4. Кросс-коэффициенты: концепция «отношения коэффициентов»

Было бы очень полезно, если бы мы могли расширить изложенные выше идеи, чтобы облегчить идентификацию наблюдаемых объектов при общих трехмерных преобразованиях. В самом деле, очевидный вопрос заключается в том, дает ли нам знание *отношений коэффициентов* (ratio of ratios) постоянный инвариант, которые обеспечивают подходящие обобщения. Ответ

заключается в том, что отношения коэффициентов обеспечивают полезные дополнительные инварианты. Сейчас мы это увидим.

Чтобы определить подходящие отношения коэффициентов расстояний, мы начнем с изучения набора из 4 коллинеарных точек на объекте. На рис. 1.21 показан такой набор из 4 точек (P_1, P_2, P_3, P_4) и их преобразование (Q_1, Q_2, Q_3, Q_4), например производимое системой формирования изображения с оптическим центром S (c, d). Выбрав подходящие пары наклонных осей, мы можем выразить координаты двух наборов точек в следующем виде:

$$(x_1, 0), (x_2, 0), (x_3, 0), (x_4, 0); \\ (0, y_1), (0, y_2), (0, y_3), (0, y_4).$$

Взяв точки P_i, Q_i , ($i = 1, \dots, 4$), можно записать отношение $CQ_i : PQ_i$ как в виде $c/-x_i$, так и в виде $(d - y_i)/y_i$. Приравнивание этих величин сразу дает:

$$\frac{c}{x_i} + \frac{d}{y_i} = 1. \quad (1.53)$$

Взяв подходящие разности, исключаяющие все абсолютные положения, и выполнив преобразования, мы в конечном итоге получаем формулу

$$\left(\frac{x_2 - x_4}{x_3 - x_4} \right) \bigg/ \left(\frac{x_2 - x_1}{x_3 - x_1} \right) = \left(\frac{y_2 - y_4}{y_3 - y_4} \right) \bigg/ \left(\frac{y_2 - y_1}{y_3 - y_1} \right). \quad (1.54)$$

Эта формула *инвариантного сложного отношения* (cross ratio) подтверждает возможность построения параметра, инвариантного к перспективным преобразованиям. В частности, 4 коллинеарные точки, рассматриваемые с любой точки зрения, дают одинаковое значение сложного отношения, определяемое как:

$$C(P_1, P_2, P_3, P_4) = \frac{(x_3 - x_1)(x_2 - x_4)}{(x_2 - x_1)(x_3 - x_4)}. \quad (1.55)$$

В дальнейшем мы будем обозначать это конкретное сложное отношение как κ . Обратите внимание, что существуют $4! = 24$ возможных способа расположения 4 коллинеарных точек на прямой линии, и, следовательно, для любого объекта возможны 24 значения сложного отношения. Однако не все они различны, и на самом деле существует только 6 различных значений: легко показать, что это κ , $1 - \kappa$, $\kappa/(\kappa - 1)$ и их обратные значения. Интересно, что нумерация точек в обратном порядке (что соответствовало бы просмотру линии с другой стороны) оставляет сложное отношение неизменным. Тем не менее неудобно, что один и тот же инвариант имеет 6 различных проявлений, так как это означает, что необходимо просмотреть 6 различных значений индекса, прежде чем можно будет идентифицировать класс объекта. С другой стороны, если точки маркируются по порядку вдоль каждой линии, а не случайным образом, можно обойти эту ситуацию.

Пока нам удалось получить только один проективный инвариант, и это соответствует довольно простому случаю четырех коллинеарных точек. Полез-

ность данной меры значительно возрастает, если заметить, что 4 коллинеарные точки, взятые вместе с другой точкой, определяют пучок параллельных компланарных линий, проходящих через последнюю точку. Ясно, что мы можем присвоить этому пучку прямых уникальное сложное отношение, равное поперечному отношению коллинеарных точек на любой линии, проходящей через них. Фактически, рассмотрев углы между различными линиями и применив правило синусов 4 раза, мы получаем следующую формулу:

$$C(P_1, P_2, P_3, P_4) = \frac{\sin \alpha_{13} \sin \alpha_{24}}{\sin \alpha_{12} \sin \alpha_{34}}. \quad (1.56)$$

Таким образом, сложное инвариантное отношение зависит только от углов пучка линий.

Мы можем расширить эту концепцию до 4 параллельных плоскостей, так как параллельные линии могут быть спроецированы на 4 параллельные плоскости после определения отдельной оси параллелизма. Поскольку таких осей бесконечно много, существует бесконечно много способов выбора наборов плоскостей. Таким образом, исходный простой результат для коллинеарных точек можно распространить на гораздо более общий случай.

В заключение отметим, что мы начали с попытки обобщить случай четырех коллинеарных точек, но в результате сначала нашли двойственную ситуацию, в которой точки становятся линиями, также описываемыми сложным отношением, а затем нашли расширение, в котором сложным отношением описываются плоскости. Теперь вернемся к случаю с четырьмя коллинеарными точками и посмотрим, как мы можем расширить его другими способами.

1.4.5. Инварианты для неколлинеарных точек

Прежде всего давайте представим, что не все точки коллинеарны: в частности, предположим, что одна точка не коллинеарна другим трем. Если это так, то для расчета сложного отношения недостаточно информации. Однако если доступна еще одна компланарная точка, мы можем провести воображаемую линию между неколлинеарными точками, чтобы пересечь линию, проходящую через другие три точки: тогда это позволит вычислить сложное отношение (рис. 1.22а). Тем не менее это далеко от общего решения задачи нахождения характеристики множества неколлинеарных точек. Мы могли бы спросить, сколько точечных признаков общего положения на плоскости потребуется для вычисления инварианта. На самом деле ответ – 5, так как тот факт, что мы можем составить сложное отношение из углов между 4 линиями, немедленно означает, что построение пучка из 4 прямых из 5 точек определяет инвариант сложного отношения (рис. 1.22b).

Хотя значение этого сложного отношения обеспечивает необходимое условие для совпадения между двумя наборами из 5 общих компланарных точек, это может быть случайным совпадением, поскольку условие зависит только от относительных направлений между различными точками и контрольной

точкой, т. е. любая из неререферентных точек определяется только относительно линии, на которой она лежит. Ясно, что два сложных отношения, образованных взятием двух опорных точек, будут однозначно определять направления всех оставшихся точек (рис. 1.22с). Интересно, что хотя в результате такого рода процедуры можно получить не менее пяти сложных отношений, оказывается, что существует только два функционально независимых сложных отношения – в основном потому, что положение любой точки определяется, когда известно ее направление относительно двух других точек.

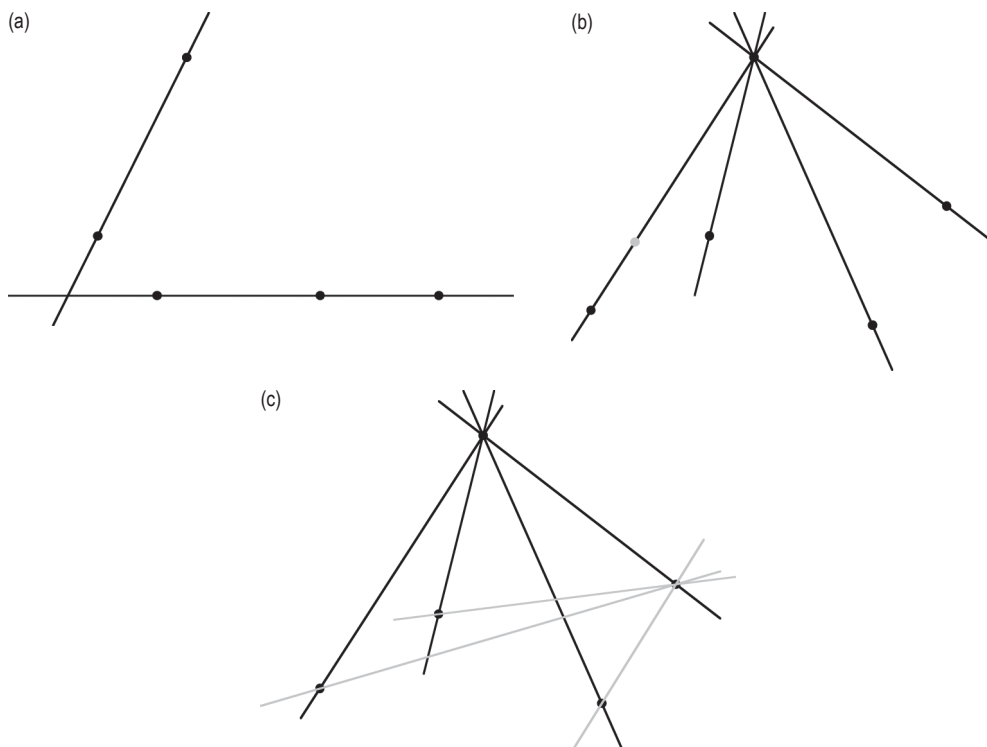


Рис. 1.22 ❖ Расчет инвариантов для набора неколлинеарных точек. На (а) показано, как добавление пятой точки к набору из четырех точек, одна из которых не коллинеарна остальным, позволяет вычислить сложное отношение; (б) показывает, как вычисление может быть распространено на любой набор неколлинеарных точек; также показана дополнительная (серая) точка, которую однократное сложное отношение не может отличить от других точек на той же линии. На (с) показано, как любая неспособность однозначно идентифицировать точку может быть преодолена путем вычисления сложного отношения второго пучка, полученного из пяти исходных точек

Заметим, что на рис. 1.22 отсутствует еще один интересный случай – ситуация с двумя точками и двумя прямыми. Построив линию, соединяющую две точки, и производя ее до пересечения с двумя линиями, мы получим 4 точки на одной линии; таким образом, конфигурация характеризуется одним

сложным отношением. Заметьте также, что две линии можно продлять до тех пор, пока они не соединятся, и дальнейшие линии могут быть построены из пересечения по двум точкам: это дает пучок линий, характеризующийся одним сложным отношением: последнее должно иметь то же значение, что и вычисленное для 4 коллинеарных точек.

Далее мы рассмотрим проблему нахождения плоскости объекта в практических ситуациях, например в случае *собственного движения* (egomotion), включая управление транспортным средством. Предположим, что от одного кадра к другому можно наблюдать набор из 4 коллинеарных точек. Если точки находятся в одной плоскости, то сложное отношение будет оставаться постоянным, но если одна из них приподнята над плоскостью земли (как в случае с неровностью на дороге), то сложное отношение будет меняться от кадра к кадру. Взяв большее количество точек, можно путем исключения определить, какие из них находятся на плоскости земли, а какие нет: обратите внимание, что все это возможно без какой-либо калибровки камеры, что является основным преимуществом использования проективных инвариантов. Впрочем, существует потенциальная проблема с нерелевантными плоскостями, такими как вертикальные грани зданий. Проверка сложного отношения очень устойчива к точке зрения и положению и просто устанавливает, компланарны ли проверяемые точки. Но, используя достаточно большое количество независимых наборов точек, можно отличить одну плоскость от другой.

1.4.6. Обнаружение точки схода

В этом разделе мы рассмотрим *точки схода* (vanishing point, VP) и способы их обнаружения. Простая привязка к точкам схода дает человеческому мозгу глубокое понимание изображения и в немалой степени помогает ему глобально интерпретировать происходящее на изображении (рис. 1.23). Следовательно, точка схода может служить отправной точкой и для машинной интерпретации изображения. Это особенно ценно в ситуации с реальными трехмерными изображениями, воплощающими все сложности полноперспективной проекции, поэтому исследователи потратили много усилий на поиск методов обнаружения и использования точек схода.

Обнаружение точек схода обычно проводят в два этапа: сначала локализуют все прямые линии на изображении, затем находят, какие из прямых проходят через общие точки, причем последние интерпретируются как точки схода. Поиск линий с помощью преобразования Хафа не должен вызывать затруднений, хотя края текстуры иногда мешают точному и обоснованному расположению линий. По сути, для обнаружения точек схода требуется второе преобразование Хафа, в котором целые линии накапливаются в пространстве параметров, что приводит к четко определенным пикам (соответствующим точкам схода, причем несколько линий перекрываются. На практике линии, которые являются объектами голосования, должны быть продлены, чтобы охватить все возможные местоположения точек схода. Эта процедура адекватна, когда точки схода располагаются в пределах исходного

пространства изображения, но часто бывает так, что они будут вне исходного изображения (рис. 1.23) и могут даже находиться в бесконечности. Это означает, что не получится использовать пространство параметров, подобное изображению, даже если оно выходит за пределы исходного пространства изображения. Другая проблема заключается в том, что для удаленных точек схода пики в пространстве параметров будут разбросаны на значительном расстоянии, поэтому чувствительность обнаружения будет слабой, а точность определения местоположения будет низкой.

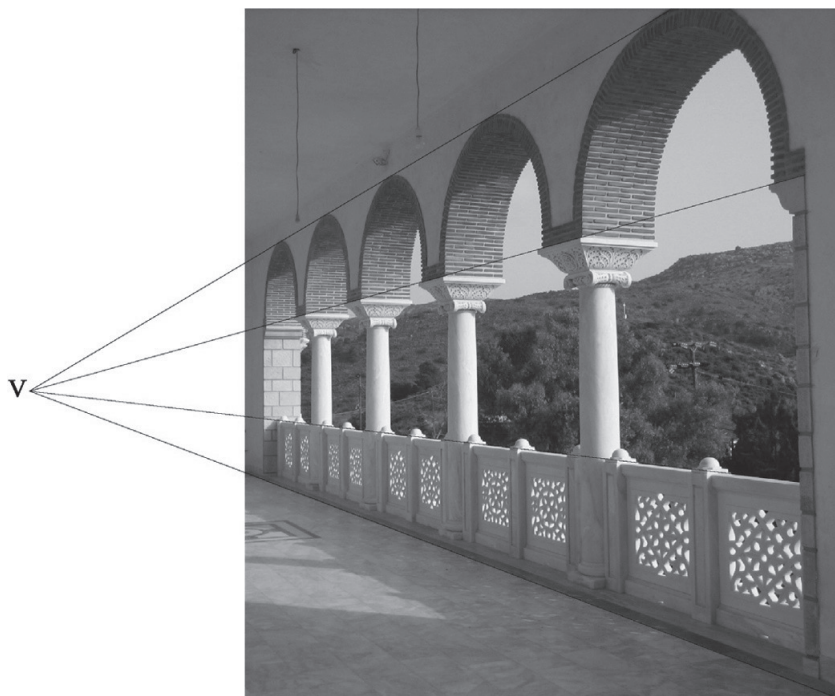


Рис. 1.23 ❖ Положение точки схода. На этом рисунке кажется, что параллельные линии на арках сходятся в точке V за пределами изображения. В общем случае точки схода могут лежать на любом расстоянии и даже находиться в бесконечности

К счастью, Маги и Аггарвал (Magee, Aggarwal, 1984) нашли улучшенное представление для поиска точек схода. Они построили единичную сферу G , называемую *сферой Гаусса*, вокруг центра проекции камеры и использовали G вместо расширенной плоскости изображения в качестве пространства параметров. В этом представлении точки схода появляются на конечных расстояниях даже в тех случаях, когда в противном случае они казались бы находящимися в бесконечности. Чтобы этот метод работал, должно быть однозначное соответствие между точками в двух представлениях, и это явно верно (заметим, что задняя половина гауссовой сферы не используется). Однако представление в виде гауссовой сферы не лишено проблем: в част-

ности, многие нерелевантные голоса будут получены линиями, которые не параллельны в реальном трехмерном пространстве (часто лишь небольшое подмножество линий на изображении проходит через точку схода). Чтобы решить эту проблему, пары линий рассматриваются по очереди, а их точки пересечения накапливаются в качестве голосов только в том случае, если считается, что линии каждой пары происходят из параллельных линий в трехмерном пространстве (например, они должны иметь совместимые градиенты в изображении). Эта процедура резко ограничивает как количество голосов, записанных в пространстве параметров, так и количество нерелевантных пиков. Тем не менее общая стоимость по-прежнему значительна, поскольку пропорциональна количеству пар линий. Таким образом, если имеется N линий, число пар равно $N C_2 = \frac{1}{2}N(N-1)$, поэтому результат равен $O(N^2)$.

Вышеупомянутая процедура важна, поскольку она обеспечивает надежные средства для выполнения поиска точек схождения и эффективного исключения изолированных линий и помех изображения. В случае движущегося робота или другой системы, оснащенной компьютерным зрением, выявление соответствия между точками схода, видимыми на последовательных изображениях, приводит к значительно большей уверенности в интерпретации каждого изображения.

1.4.7. Подробнее о точках схода

Одним из преимуществ инварианта сложного отношения является то, что он может появляться во многих ситуациях и в каждом случае давать еще один точный результат. Интересным примером является сценарий, когда дорога или тротуар имеет каменные плиты, границы которых хорошо очерчены и легко измеримы. Их можно использовать для оценки положения точки схода на плоскости земли. Представьте, что вы смотрите на каменные плиты под углом сверху, а камера или глаза выровнены горизонтально. Тогда у нас есть геометрическая структура, аналогичная рис. 1.24, где точки O , H_1 , H_2 лежат на плоскости земли, а O , V_1 , V_2 , V_3 находятся в плоскости изображения.

Если мы возьмем S в качестве центра проекции, то сложное отношение, образованное точками O , V_1 , V_2 , V_3 , должно иметь такое же значение, что и сложное отношение, образованное точками O , H_1 , H_2 и бесконечностью в горизонтальном направлении. Предположим, что OH_1 и H_1H_2 имеют известные длины a и b ; приравнивание значений сложного отношения дает:

$$\frac{y_1(y_3 - y_2)}{y_2(y_3 - y_1)} = \frac{x_1}{x_2} = \frac{a}{a + b}. \quad (1.57)$$

(Обратите внимание, что на рис. 1.24 значения y отсчитываются от O , а не от V_3 .) Это позволяет нам найти y_3 . Принимая $a = b$ (как, например, в случае с каменными плитами), мы находим, что:

$$y_3 = \frac{y_1 y_2}{2y_1 - y_2}. \quad (1.58)$$

Найдя y_3 , мы вычислили направление точки схода независимо от того, горизонтальна ли плоскость земли, на которой она лежит, и горизонтальна ли ось камеры. Заметим, что это доказательство на самом деле не предполагает, что точки V_1, V_2, V_3 находятся вертикально над началом координат или что линия OH_1H_2 горизонтальна; мы лишь полагаем, что эти точки лежат на двух компланарных прямых и что C находится в той же плоскости.

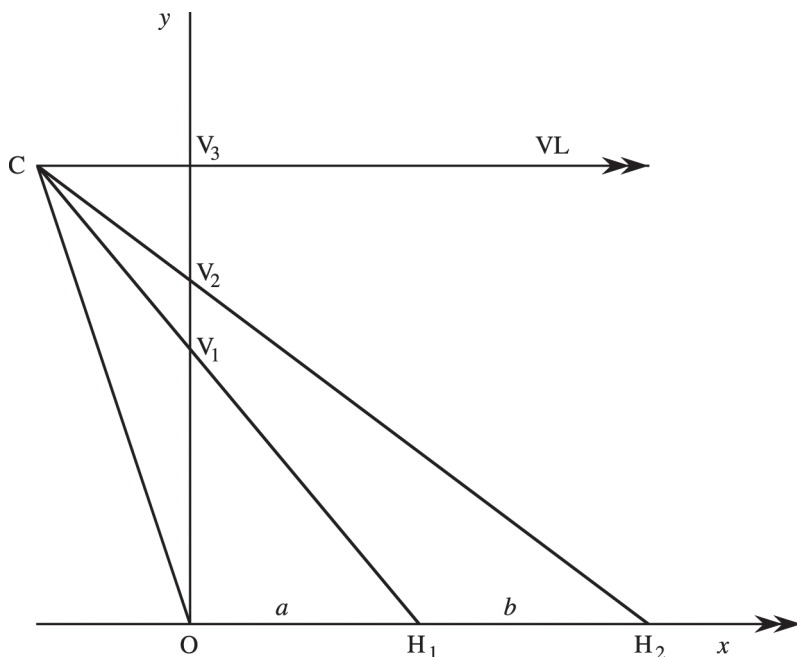


Рис. 1.24 ❖ Схема нахождения линии схода по известной паре интервалов: C – центр проекции, VL – направление линии схода, параллельное плоскости земли OH_1H_2 . Хотя плоскость камеры $OV_1V_2V_3$ нарисована перпендикулярно плоскости земли, это не обязательно для успешной работы алгоритма (см. текст выше)

1.4.8. Промежуточный итог: значение инвариантов

Разделы 1.4.2–1.4.6 были посвящены тому, чтобы дать некоторое представление о важном понятии инвариантов и их применении в распознавании изображений. Тема получает значительное развитие, когда рассматриваются отношения коэффициентов расстояний, и эта идея естественным образом приводит к сложному инвариантному отношению. Хотя его первоначальное применение заключается в распознавании расстояний между точками на линии, оно непосредственно обобщается на угловые расстояния для пучков линий, а также на угловое расстояние между плоскостями. Дальнейшим развитием идеи является разработка инвариантов, которые могут описывать

наборы неколлинеарных точек, и двух сложных отношений достаточно, чтобы охарактеризовать набор из 5 неколлинеарных точек на плоскости.

Существует много других теорем и типов инвариантов, но из-за недостатка места мы вынуждены ограничиться лишь их упоминанием. В процессе развития примеров точек и линий, обсуждавшихся выше, были созданы инварианты, включающие кривые второго порядка (*коники*, *conic*): коника и две компланарные некасательные прямые, коника и две компланарные точки, две компланарные коники. В целом ценность инвариантов заключается в выполнении эффективных с вычислительной точки зрения проверок того, могут ли точки или другие признаки принадлежать конкретным объектам. Кроме того, они достигают этого без необходимости калибровки камеры или знания точки обзора камеры (хотя существует неявное предположение, что камера является евклидовой).

1.4.9. Преобразование изображения для калибровки камеры

Когда изображения получаются из трехмерных сцен, точное положение и ориентация камеры часто неизвестны, и необходимо связать их с некоторой глобальной системой отсчета. Это особенно важно, если необходимо производить точные измерения объектов по их изображениям, например в приложениях визуального инспектирования. С другой стороны, иногда можно обойтись без такой подробной информации – как в случае стационарной охранной системы обнаружения злоумышленников или системы подсчета автомобилей на автомагистрали. Есть и более сложные случаи, например когда камеры можно вращать или перемещать на манипуляторе робота или исследуемые объекты могут свободно перемещаться в пространстве. В таких случаях главной задачей становится «внешняя», а также «внутренняя» *калибровка камеры* (полное объяснение этих терминов см. в разделе 1.4.11).

Прежде чем мы сможем рассмотреть калибровку камеры, нам нужно детально разобрать преобразования, которые могут происходить между исходными *мировыми точками*¹ и формированием конечного изображения. В частности, мы рассматриваем повороты и перемещения точек объекта относительно глобальной системы отсчета. После поворота на угол θ вокруг оси Z (рис. 1.25) координаты общей точки (X, Y) меняются на:

$$X' = X \cos \theta - Y \sin \theta; \quad (1.59)$$

$$Y' = X \sin \theta + Y \cos \theta. \quad (1.60)$$

Далее мы обобщаем этот результат на трехмерное пространство и выражаем его как матрицу поворота θ вокруг оси Z :

¹ Точка пространства событий. – Прим. перев.

$$\mathbf{Z}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.61)$$

Аналогичные матрицы применяются для поворотов ψ вокруг оси X и ϕ вокруг оси Y . Применяя последовательности таких поворотов, мы получаем следующий общий результат, выражающий произвольное трехмерное вращение \mathbf{R} :

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (1.62)$$

Обратите внимание, что матрица вращения \mathbf{R} не является полностью общей: она ортогональна и, таким образом, обладает тем свойством, что $\mathbf{R}^{-1} = \mathbf{R}^T$.

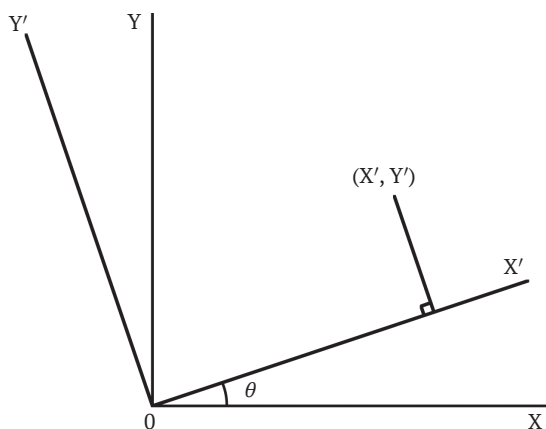


Рис. 1.25 ❖ Эффект вращения θ вокруг начала координат

В отличие от вращения, перенос на расстояние (T_1, T_2, T_3) определяется следующим уравнением:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} Z \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}, \quad (1.63)$$

которое не выражается через мультипликативную матрицу 3×3 . Чтобы объединить повороты и переносы в общую мультипликативную формулу, мы должны использовать *однородные координаты*. Для этого матрицы должны быть увеличены до 4×4 , а требуемое преобразование должно иметь вид:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (1.64)$$

Эта форма является достаточно общей, чтобы включать масштабирование по размеру объекта, а также преобразования в виде сдвига и наклона.

Во всех рассмотренных выше случаях можно заметить, что нижняя строка обобщенной матрицы перемещений является избыточной. На самом деле мы можем найти хорошее применение этому ряду в некоторых других типах преобразования. Особый интерес в этом контексте представляет случай перспективной проекции. В соответствии с разделом 1.4.1 уравнения для проекции точек объекта в точки изображения имеют вид:

$$x = f X/Z; \quad (1.65)$$

$$y = f Y/Z; \quad (1.66)$$

$$z = f. \quad (1.67)$$

Чтобы включить перспективную проекцию в приведенные выше формулы, нам нужно изучить преобразование однородных координат:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ Z/f \end{bmatrix}. \quad (1.68)$$

Ключом к пониманию этого преобразования является то, что деление на четвертую координату дает требуемые значения преобразованных декартовых координат ($f X/Z, f Y/Z, f$).

Давайте теперь присмотримся к этому результату. Во-первых, мы нашли матричное преобразование 4×4 , которое работает с однородными четырехмерными координатами. Они не соответствуют напрямую реальным координатам, но из них можно вычислить реальные трехмерные координаты, разделив первые три на четвертую однородную координату. Таким образом, однородные координаты произвольны в том смысле, что все они могут быть умножены на один и тот же постоянный множитель без каких-либо изменений в окончательной интерпретации.

Преимуществом использования однородных координат является удобство использования единой мультипликативной матрицы для любого преобразования, несмотря на то что перспективные преобразования по своей сути нелинейны: таким образом, довольно сложное нелинейное преобразование может быть сведено к более простому линейному преобразованию. Это упрощает компьютерный расчет преобразований координат объекта и другие вычисления, например для калибровки камеры (см. ниже). Заметим также, что почти каждое преобразование можно обратить, обратив соответствующую

однородную матрицу преобразования. Исключением является преобразование перспективы, для которого фиксированное значение z приводит к тому, что Z просто неизвестно, а X , Y известны только относительно значения Z (отсюда вытекает необходимость бинокулярного зрения или других средств определения глубины сцены).

1.4.10. Калибровка камеры

Выше мы рассмотрели, как однородные системы координат образуют удобное линейное матричное представление 4×4 для трехмерных преобразований, включая перемещение и вращение твердого тела, а также нежестких операций, включая масштабирование, наклон и перспективную проекцию. В этом последнем случае неявно предполагалось, что системы координат камеры и мира идентичны, поскольку координаты изображения были выражены в одной и той же системе отсчета. Однако в общем случае объекты, видимые камерой, будут иметь положения, которые могут быть известны в мировых координатах, но которые не будут известны *априори* в координатах камеры, поскольку камера в общем случае будет установлена в произвольном положении и будет «смотреть» в произвольном направлении. Следовательно, система камеры должна быть откалибрована, прежде чем изображения можно будет использовать для практических приложений, таких как роботизированная сборка или размещение предметов. Полезный подход состоит в том, чтобы предположить существование некоего обобщенного преобразования между мировыми координатами и изображением, видимым камерой при перспективной проекции, и разместить на изображении различные калибровочные точки, которые были размещены в известных местах сцены. При наличии достаточного количества таких точек должна появиться возможность вычислить параметры преобразования, а затем все точки изображения могут быть точно интерпретированы до тех пор, пока не потребуются повторная калибровка.

В дальнейшем мы обнаружим, что существует два типа калибровки камеры: первый – это *внешняя калибровка* (extrinsic calibration), при которой положение камеры определяется относительно мировых координат через ее внешние параметры; вторая – *внутренняя калибровка* (intrinsic calibration), при которой положение изображения (и пикселя) определяется в зависимости от внутренних параметров камеры. Важными факторами, которые мы должны обсудить, являются количество внешних и внутренних параметров и их геометрическое значение.

Прежде всего нам нужно записать математическую формулу в общем виде, используя общее однородное преобразование G , которое принимает вид:

$$\begin{bmatrix} X_H \\ Y_H \\ Z_H \\ H \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \\ G_{41} & G_{42} & G_{43} & G_{44} \end{bmatrix} \begin{bmatrix} Z \\ Y \\ X \\ 1 \end{bmatrix}. \quad (1.69)$$

Обратите внимание, что окончательные декартовы координаты, появляющиеся на изображении, равны $(x, y, z) = (x, y, f)$, и они вычисляются из первых трех однородных координат путем деления на четвертую:

$$x = X_H/H = (G_{11}X + G_{12}Y + G_{13}Z + G_{14})/(G_{41}X + G_{42}Y + G_{43}Z + G_{44}); \quad (1.70)$$

$$y = Y_H/H = (G_{21}X + G_{22}Y + G_{23}Z + G_{24})/(G_{41}X + G_{42}Y + G_{43}Z + G_{44}); \quad (1.71)$$

$$z = Z_H/H = (G_{31}X + G_{32}Y + G_{33}Z + G_{34})/(G_{41}X + G_{42}Y + G_{43}Z + G_{44}). \quad (1.72)$$

Однако, поскольку мы знаем z , нет смысла определять параметры G_{31} , G_{32} , G_{33} , G_{34} . Следовательно, мы можем перейти к нахождению остальных параметров. На самом деле, поскольку имеют смысл только отношения однородных координат, необходимо вычислять лишь отношения значений G_{ij} , и обычно G_{44} принимают за единицу: остается определить только 11 параметров. Перемножая первые два уравнения и выполняя преобразования, получаем:

$$G_{11}X + G_{12}Y + G_{13}Z + G_{14} - x(G_{41}X + G_{42}Y + G_{43}Z) = x; \quad (1.73)$$

$$G_{21}X + G_{22}Y + G_{23}Z + G_{24} - y(G_{41}X + G_{42}Y + G_{43}Z) = y. \quad (1.74)$$

Понимание того факта, что единственная мировая точка (X, Y, Z) , которая, как известно, соответствует точке изображения (x, y) , дает нам два уравнения приведенной выше формы; требуется минимум 6 таких точек, чтобы обеспечить значения для всех 11 параметров G_{ij} . Важным фактором является то, что мировые точки, используемые для расчета, должны приводить к независимым уравнениям, поэтому важно, чтобы они не были компланарными. Точнее, должно быть не менее 6 точек, никакие четыре из которых не лежат в одной плоскости. Однако дополнительные точки полезны тем, что приводят к переопределению параметров и повышают точность их вычисления. Нет никаких причин, по которым дополнительные точки не должны лежать в одной плоскости с существующими точками: действительно, обычный подход состоит в повороте куба таким образом, чтобы были видны три его грани, причем каждая грань имеет паттерн из квадратов с 30–40 легко различимыми признаками угла.

Для вычисления 11 параметров можно использовать метод наименьших квадратов. Во-первых, $2n$ уравнений (для n точек) должны быть выражены в матричной форме:

$$A\mathbf{g} = \boldsymbol{\xi}, \quad (1.75)$$

где A – матрица коэффициентов $2n \times 11$, которая перемножается с G -матрицей, теперь имеющей вид

$$\mathbf{g} = (G_{11}G_{12}G_{13}G_{14}G_{21}G_{22}G_{23}G_{24}G_{41}G_{42}G_{43})^T, \quad (1.76)$$

а $\boldsymbol{\xi}$ представляет собой $2n$ -элементный вектор-столбец координат изображения. Псевдообратное решение имеет вид:

$$\mathbf{g} = A^\dagger \boldsymbol{\xi}, \quad (1.77)$$

где

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T. \quad (1.78)$$

1.4.11. Внутренние и внешние параметры

На этом этапе полезно более подробно рассмотреть общее преобразование, ведущее к калибровке камеры. Когда мы калибруем камеру, мы на самом деле пытаемся совместить камеру и мировые системы координат. Первый шаг – переместить начало мировой системы координат в начало системы координат камеры. Второй шаг – повернуть мировую систему координат, пока ее оси не совпадут с осями системы координат камеры. Третий шаг – смещение плоскости изображения вбок до полного совпадения двух систем координат (этот шаг необходим, поскольку изначально неизвестно, какая точка мировой системы координат соответствует главной точке изображения).

Во время этого процесса следует помнить об одном важном моменте. Если координаты камеры заданы как \mathbf{C} , то сдвиг \mathbf{T} , необходимый на первом шаге, будет равен $-\mathbf{C}$. Точно так же требуемые повороты будут обратными тем, которые соответствуют фактическим ориентациям камеры. Причина этих разворотов в том, что (например) вращение объекта (в данном случае камеры) вперед дает тот же эффект, что и вращение осей назад. Таким образом, все операции должны выполняться с аргументами, обратными указанным выше в разделе 1.4.1. Таким образом, полное преобразование для калибровки камеры будет следующим:

$$\mathbf{G} = \mathbf{PLRT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_1 \\ 0 & 1 & 0 & T_2 \\ 0 & 0 & 1 & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.79)$$

где матрица \mathbf{P} учитывает преобразование перспективы, необходимое для формирования изображения. На самом деле обычно преобразования \mathbf{P} и \mathbf{L} группируются вместе и называются преобразованиями внутренней камеры, которые включают *внутренние параметры камеры*, в то время как \mathbf{R} и \mathbf{T} берутся вместе как преобразования внешней камеры, соответствующие *внешним параметрам камеры*. Следовательно:

$$\mathbf{G}_{\text{внутр}} = \mathbf{PL} = \begin{bmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 1/f & t_3/f \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1/f \end{bmatrix}. \quad (1.80)$$

В матрице для $\mathbf{G}_{\text{внутр}}$ мы предположили, что исходная матрица переноса \mathbf{T} перемещает центр проекции камеры в правильное положение, так что значение t_3 можно сделать равным нулю, оставляя матрицу 3×3 .

Хотя приведенная выше трактовка дает хорошее представление о скрытом значении G , она не является общей, поскольку мы до сих пор не включали параметры масштабирования и перекоса во внутреннюю матрицу. На самом деле обобщенная форма $G_{\text{внутр}}$ выглядит так:

$$G_{\text{внутр}} = \begin{bmatrix} s_1 & b_1 & t_1 \\ b_2 & s_2 & t_2 \\ 0 & 0 & 1/f \end{bmatrix}. \quad (1.81)$$

Потенциально $G_{\text{внутр}}$ должна включать преобразования для исправления (1) ошибок масштабирования, (2) ошибок переноса, (3) ошибок наклона датчика, (4) ошибок сдвига датчика, (5) неизвестной ориентации датчика в плоскости изображения. Очевидно, что ошибки переноса исправляются путем корректировки t_1 и t_2 . Все остальные настройки связаны со значениями подматрицы 2×2 , содержащей параметры s_1, s_2, b_1, b_2 .

Однако обратите внимание, что применение этой матрицы выполняет вращение в плоскости изображения сразу после того, как $G_{\text{наруж}}$ выполнила вращение в мировых координатах, и разделить два вращения практически невозможно. Это объясняет, почему теперь у нас есть в общей сложности 6 внешних и 6 внутренних параметров – всего 12, а не ожидаемые 11. В результате лучше исключить пункт 5 из приведенного выше списка внутренних переносов и отнести его к внешним параметрам. Поскольку вращательная составляющая в $G_{\text{внутр}}$ была исключена, b_1 и b_2 теперь должны быть равны, а внутренние параметры будут следующими: s_1, s_2, b, t_1, t_2 . Обратите внимание, что коэффициент $1/f$ обеспечивает масштабирование, которое нельзя отделить от других коэффициентов масштабирования во время калибровки камеры без специального (то есть отдельного) измерения f . Таким образом, у нас есть в общей сложности 6 параметров от $G_{\text{наруж}}$ и 5 параметров от $G_{\text{внутр}}$: всего 11, что равно числу, указанному в предыдущем разделе.

1.4.12. Многогракурсное зрение

В течение 1990-х гг. был достигнут значительный прогресс в трехмерном зрении путем изучения того, какую информацию можно извлечь из изображений с некалиброванных камер, рассматривающих мир с разных ракурсов. На первый взгляд, если вспомнить об усилиях, которые мы предприняли в предыдущих разделах этой главы, чтобы понять, как именно следует калибровать камеры, это может показаться бессмысленным. Тем не менее в изучении *многогракурсного зрения* (multiple view vision) есть значительная потенциальная выгода – не в последнюю очередь потому, что нам доступны тысячи часов видео, снятых некалиброванными камерами, в том числе применяемых в видеонаблюдении и в киноиндустрии. В таких случаях необходимо максимально использовать имеющийся материал. Однако потребность намного шире. Существует множество ситуаций, в которых параметры камеры могут изменяться из-за колебаний температуры или из-за того, что настройки масштабирования или фокусировки были скорректированы; оче-

видно, что практически невозможно по каждому поводу перекалибровывать камеру с использованием точно изготовленных тестовых объектов. Наконец, если используется несколько камер, каждую придется калибровать отдельно, а результаты сравнивать, чтобы свести к минимуму ошибку комбинирования. Гораздо лучше исследовать систему в целом и калибровать ее на реальных просматриваемых сценах.

На самом деле мы уже встречали некоторые аспекты этого подхода в виде инвариантов, последовательно получаемых одной камерой. Например, если просматривается серия из 4 коллинеарных точек и проверяется их поперечное соотношение, будет обнаружено, что оно остается постоянным, по мере того как камера движется вперед, меняет ориентацию или рассматривает точки все более наклонно – до тех пор, пока все они остаются в пределах поля зрения. В этом случае все, что требуется для выполнения распознавания и поддержания осведомленности об объекте, – это некалиброванная, но не искажающая камера.

Чтобы понять, как можно интерпретировать изображения в более широком смысле, используя несколько видов – будь то с одной и той же камеры, перемещенной в разные места, или с нескольких камер с перекрывающимися видами мира, – нам нужно вернуться к основам и более тщательно изучить такие понятия, как бинокулярное зрение и эпиполярные ограничения. В частности, будут задействованы две важные матрицы – *существенная* (essential matrix) и *фундаментальная* (fundamental matrix). Мы начнем с существенной матрицы, а затем обобщим эту идею на фундаментальную. Но сначала нам нужно рассмотреть геометрию системы двух камер с общим ракурсом.

1.4.13. Обобщенная геометрия стереозрения

В разделе 1.4.1 мы рассмотрели проблему стереосоответствия и уже упростили задачу, выбрав две камеры, плоскости изображения которых были не только параллельны, но и находились в одной плоскости. Этот подход сделал геометрию восприятия глубины особенно простой, но никак не использовал возможности *зрительной системы человека* (human visual system, HVS) иметь ненулевой угол расхождения между двумя изображениями.

Здесь мы обобщаем ситуацию, чтобы охватить возможность несоответствия в сочетании с существенным расхождением. На рис. 1.26 показана измененная геометрическая схема. Сначала обратите внимание, что наблюдение реальной точки P сцены дает точки P_1 и P_2 на двух изображениях; что P_1 может соответствовать любой точке эпиполярной линии E_2 на изображении 2; и точно так же эта точка P_2 может соответствовать любой точке эпиполярной линии E_1 на изображении 1. В самом деле, так называемая *эпиполярная плоскость* P – это плоскость, содержащая P и точки проекций C_1 и C_2 двух камер: эпиполярные линии (раздел 1.4.1), таким образом, являются прямыми линиями, по которым эта плоскость пересекает две плоскости изображения. Кроме того, линия, соединяющая C_1 и C_2 , пересекает плоскости изображения в так называемых *эпиполюсах* (epipole) e_1 и e_2 : их можно рассматривать как

изображения альтернативных точек проекции камеры. Обратите внимание, что все эпиллярные плоскости проходят через точки C_1 , C_2 и e_1 , e_2 : это означает, что все эпиллярные линии на двух изображениях проходят через соответствующие эпиллюсы.

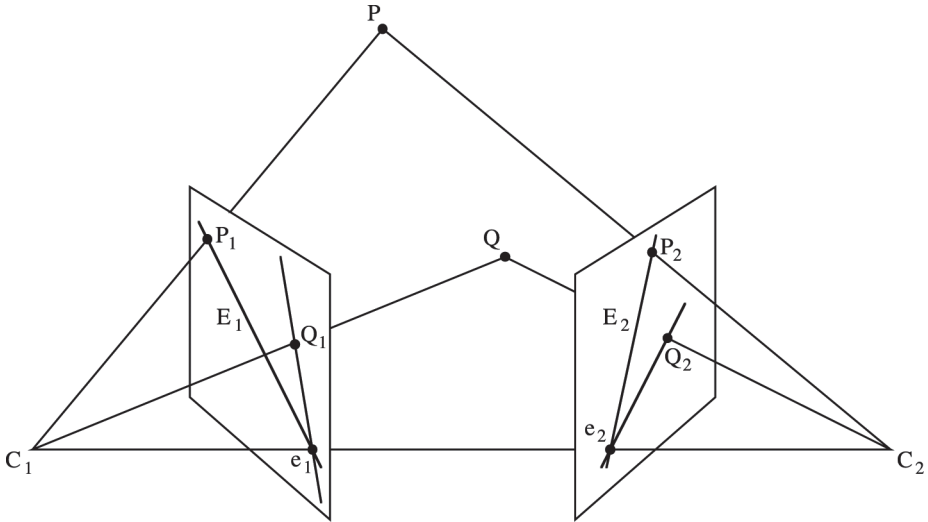


Рис. 1.26 ❖ Обобщенное изображение сцены, наблюдаемой с двух точек зрения. В этом случае имеется существенная вергенция. Все эпиллярные линии на левом изображении проходят через эпиллюс e_1 : из них показана только линия E_1 . То же самое можно сказать и о правом изображении

1.4.14. Существенная матрица

В этом разделе мы начнем с векторов P_1, P_2 , направленных из C_1, C_2 в P , а также вектора C , исходящего из C_1 в C_2 . Вычитание векторов дает:

$$P_2 = P_1 - C. \quad (1.82)$$

Мы также знаем, что P_1, P_2 и C компланарны, и условие компланарности следующее:

$$P_2 \cdot C \times P_1 = 0. \quad (1.83)$$

Чтобы пойти дальше, нам нужно связать векторы P_1 и P_2 , когда они выражены относительно их собственных систем отсчета. Если мы возьмем эти векторы как определенные в системе отсчета C_1 , мы теперь повторно выразим P_2 в его собственной (C_2) системе отсчета, применяя перенос C и поворот координат, выраженный в виде ортогональной матрицы R . Это приводит к следующему уравнению:

$$P'_2 = RP_2 = R(P_1 - C), \quad (1.84)$$

так что

$$\mathbf{P}_2 = R^{-1}\mathbf{P}'_2 = R^T\mathbf{P}'_2. \quad (1.85)$$

Подстановка в условие компланарности дает:

$$(R^T\mathbf{P}'_2) \cdot \mathbf{C} \times \mathbf{P}_1 = 0. \quad (1.86)$$

На этом этапе полезно заменить обозначение векторного произведения, используя \mathbf{C}_\times для обозначения кососимметричной матрицы \mathbf{C}_\times , где

$$\mathbf{C}_\times = \begin{bmatrix} 0 & -C_z & C_y \\ C_z & 0 & -C_x \\ -C_y & C_x & 0 \end{bmatrix}. \quad (1.87)$$

В то же время мы следим за правильной матричной формулировкой всех векторов, соответствующим образом транспонируя. Теперь мы находим, что:

$$(R^T\mathbf{P}'_2)^T \mathbf{C}_\times \mathbf{P}_1 = 0; \quad (1.88)$$

$$\therefore \mathbf{P}'_2{}^T \mathbf{R} \mathbf{C}_\times \mathbf{P}_1 = 0. \quad (1.89)$$

Наконец, мы получаем формальное представление существенной матрицы:

$$\mathbf{P}'_2{}^T \mathbf{E} \mathbf{P}_1 = 0, \quad (1.90)$$

где существенная матрица может быть найдена как

$$\mathbf{E} = \mathbf{R} \mathbf{C}_\times. \quad (1.91)$$

Уравнение (1.90) действительно является искомым результатом: оно выражает отношение между наблюдаемыми положениями одной и той же точки в системе отсчета двух камер. Кроме того, оно сразу приводит к формулам для эпиполярных линий. Чтобы убедиться в этом, сначала обратите внимание, что координаты пикселя в кадре камеры \mathbf{C}_1 :

$$\mathbf{p}_1 = (f_1/Z_1)\mathbf{P}_1, \quad (1.92)$$

в то время как они же в кадре камеры \mathbf{C}_2 (и выраженном в членах этой системы отсчета):

$$\mathbf{p}'_2 = (f_2/Z_2)\mathbf{P}'_2. \quad (1.93)$$

Исключая \mathbf{P}_1 и \mathbf{P}'_2 и отбрасывая штрих (поскольку в перспективных плоскостях изображения чисел 1 и 2 достаточно для однозначного указания координат), мы находим:

$$\mathbf{p}_2{}^T \mathbf{E} \mathbf{p}_1 = 0, \quad (1.94)$$

так как Z_1, Z_2 и f_1, f_2 можно сократить.

Заметим теперь, что представления $\mathbf{p}_2^T E = \mathbf{l}_1^T$ и $\mathbf{l}_2 = E \mathbf{p}_1$ приводят нас к следующим соотношениям:

$$\mathbf{p}_1^T \mathbf{l}_1 = 0; \quad (1.95)$$

$$\mathbf{p}_2^T \mathbf{l}_2 = 0. \quad (1.96)$$

Это означает, что $\mathbf{l}_2 = E \mathbf{p}_1$ и $\mathbf{l}_1 = E^T \mathbf{p}_2$ являются эпполярными линиями, соответствующими точкам \mathbf{p}_1 и \mathbf{p}_2 соответственно.

1.4.15. Фундаментальная матрица

Обратите внимание, что в последней части основного вычисления матрицы мы неявно предполагали, что камеры правильно откалиброваны. В частности, \mathbf{p}_1 и \mathbf{p}_2 являются скорректированными (откалиброванными) координатами пикселя изображения. Однако нам необходимо работать с неоткалиброванными изображениями, используя необработанные измерения координат пикселей – в силу причин, указанных в разделе 1.4.12. Применяя внутренние матрицы камеры G_1 , G_2 к калиброванным координатам изображения (раздел 1.4.10), мы получаем необработанные координаты изображения:

$$\mathbf{q}_1 = G_1 \mathbf{p}_1; \quad (1.97)$$

$$\mathbf{q}_2 = G_2 \mathbf{p}_2. \quad (1.98)$$

На самом деле нам здесь нужно идти в обратном направлении, поэтому воспользуемся обратными уравнениями:

$$\mathbf{p}_1 = G_1^{-1} \mathbf{q}_1; \quad (1.99)$$

$$\mathbf{p}_2 = G_2^{-1} \mathbf{q}_2. \quad (1.100)$$

Подставив значения \mathbf{p}_1 и \mathbf{p}_2 в уравнение (1.94), находим искомое уравнение, связывающее необработанные координаты пикселя:

$$\mathbf{q}_2^T (G_2^{-1})^T E G_1^{-1} \mathbf{q}_1 = 0, \quad (1.101)$$

что может быть выражено как

$$\mathbf{q}_2^T F \mathbf{q}_1 = 0, \quad (1.102)$$

где

$$F = (G_2^{-1})^T E G_1^{-1}. \quad (1.103)$$

Матрица F называется *фундаментальной*. Поскольку она включает в себя всю информацию, которая потребуется для калибровки камер, она содержит больше свободных параметров, чем существенная матрица. Однако в других отношениях две матрицы предназначены для передачи одной и той же базовой информации, что подтверждается сходством между уравнениями (1.90) и (1.102).

1.4.16. Свойства существенной и фундаментальной матриц

Далее мы рассмотрим композицию существенной и фундаментальной матриц. В частности, обратите внимание, что C_x является множителем E , а также, косвенно, F . На самом деле они однородны по C_x , поэтому масштаб C не будет иметь значения для двух матричных уравнений (1.90) и (1.102), важно только *направление* C : масштабы E и F не существенны, и поэтому важны только относительные значения их коэффициентов. Это означает, что в E и F имеется не более 8 независимых коэффициентов. Фактически в случае F их только 7, так как C_x кососимметрична, и это гарантирует, что она имеет ранг 2, а не ранг 3 – свойство, которое передается F . Аналогичные рассуждения применимы к E , но более низкая сложность E означает, что она имеет только 5 свободных параметров. В последнем случае легко понять, что они из себя представляют: они возникают из исходных 3 параметров перемещения (\mathbf{C}) и 3 параметров вращения (\mathbf{R}), за вычетом одного параметра, соответствующего масштабу.

В этом контексте обратите внимание, что если \mathbf{C} возникает в результате переноса одной камеры, то одна и та же существенная матрица будет результатом любого масштаба \mathbf{C} : на самом деле имеет значение только направление \mathbf{C} , и те же самые эпиполярные линии будут результатом продолжающегося движения в одном и том же направлении. Фактически в этом случае мы можем интерпретировать эпиполусы как очаги расширения или сжатия. Это подчеркивает мощь данной формулировки: в частности, она рассматривает движение и смещение как единое целое.

Наконец, мы должны понять, почему в фундаментальной матрице 7 свободных параметров. Ответ относительно прост. Для каждого эпиполуса требуется 2 параметра. Кроме того, 3 параметра необходимы для сопоставления любых трех эпиполярных линий с одного изображения на другое. Но почему для сопоставления нужны только 3 эпиполярные линии? Это связано с тем, что семейство эпиполярных линий представляет собой пучок, ориентация которого связана поперечными отношениями, поэтому, зная три эпиполярные линии, можно вывести сопоставление любой другой.

1.4.17. Расчет фундаментальной матрицы

В предыдущем разделе мы показали, что фундаментальная матрица имеет 7 свободных параметров. Это означает, что должна быть возможность найти ее, идентифицируя одни и те же 7 признаков на двух изображениях. Однако, хотя это математически возможно в принципе и подходящий нелинейный алгоритм существует (Faugeras et al., 1992), было показано, что этот алгоритм может быть численно нестабильным. По сути, шум действует как дополнительная переменная, увеличивающая эффективное число степеней свободы в задаче до 8. Однако для решения этой задачи был разработан линейный алгоритм, называемый *8-точечным алгоритмом*. Любопытно, что

этот алгоритм был предложен много лет назад Лонге-Хиггинсом (Longuet-Higgins, 1981) для оценки основной матрицы, но он проявил себя на практике позже, когда Хартли (Hartley, 1995) показал, как ограничить ошибки, сначала нормализовав значения. Кроме того, используя более 8 точек, можно добиться повышенной точности, но тогда необходимо найти подходящий алгоритм, способный справиться с теперь уже переопределенными параметрами. Для этого можно использовать анализ главных компонент, подходящей процедурой которого является *разложение по сингулярным значениям матрицы* (singular value decomposition, SVD).

1.4.18. Усовершенствованные методы триангуляции

В течение нескольких лет оставалась нерешенной задача нахождения точных численных значений фундаментальной матрицы, поскольку недостаточная устойчивость связана с тем, что решение методом наименьших квадратов не справляется, когда данные искажены шумом. Эта проблема возникает всякий раз, когда шум содержит выбросы. В частности, выбросы могут возникать, когда шум препятствует встрече соответствующих линий обзора в трехмерной сцене (т. е. когда две линии обзора смещены). Очевидным и хорошо проверенным решением этой проблемы является выбор средней точки общего перпендикуляра к двум линиям обзора в качестве точки пересечения. Однако этот метод не дает оптимальных результатов в конечном счете потому, что понятия «общий перпендикуляр» и «средняя точка» математически недействительны для FPP. Впоследствии Канатани (Kanatani, 1996) смог предложить новый способ нахождения оптимальной поправки, взяв за точку пересечения такую точку, в которой суммарная величина смещения на двух плоскостях изображения минимальна. Хотя Хартли и Штурм (Hartley, Sturm, 1994) предложили похожий метод, было обнаружено, что метод Канатани на несколько порядков быстрее и не страдает эпполусными сингулярностями, возникающими при использовании метода Хартли–Штурма (Torr, Zisserman, 1997). Впоследствии, вплоть до 2019 г., продолжали появляться интересные улучшения метода. В частности, Ли и Чивера (Lee, Civera, 2019) предложили модифицированный метод средней точки – метод обобщенной взвешенной средней точки, – в котором не предполагается, что две начальные точки лежат на общем перпендикуляре. Они показали, что, хотя их метод теоретически не является оптимальным в смысле сведения к минимуму геометрических или алгебраических ошибок, он превосходит существующие методы с точки зрения скорости, простоты и комбинированной точности 2D, 3D и параллакса.

Фати (Fathy et al., 2011) попыток ситуацию следующим образом: 8-точечный алгоритм – это одношаговый метод, который обычно применяется после удаления выбросов для получения начальной оценки фундаментальной матрицы; затем он итеративно уточняется для получения более точного решения.

1.4.19. Достижения и ограничения многоракурсного зрения

В нескольких предыдущих разделах обсуждались преобразования, необходимые для калибровки камеры, и описывалось, как можно выполнить калибровку. Параметры камеры были классифицированы как «внутренние» и «внешние», что упрощает концептуальную проблему и проливает свет на источники ошибок в системе. Показано, что для проведения калибровки в общем случае, когда задействовано 11 параметров преобразования, требуется минимум 6 точек. Тем не менее, как правило, желательно увеличить количество точек, используемых для калибровки, насколько это возможно, поскольку в результате процесса усреднения может быть получен значительный выигрыш в точности.

В разделе 1.4.12 представлено многоракурсное зрение. Было показано, что эта важная тема опирается на обобщенную эпиполярную схему и приводит к идее существенных и фундаментальных матриц, которые связывают наблюдаемые положения любой точки в двух системах отсчета камеры. Была подчеркнута важность 8-точечного алгоритма для расчета любой из этих матриц – в особенности фундаментальной матрицы, которая актуальна, когда камеры не откалиброваны. Кроме того, вопросы точности и надежности при расчете фундаментальной матрицы все еще остаются предметами исследований, хотя в последние годы были достигнуты большие успехи (см. раздел 1.4.18) в отношении этапа удаления выбросов при расчете фундаментальной матрицы.

1.5. Часть D. Отслеживание движущихся объектов

1.5.1. Основные принципы отслеживания

В последние годы было разработано множество алгоритмов для интерпретации отдельных изображений и идентификации большого количества объектов на них. После этого успеха ученые обратили внимание на анализ последовательностей изображений и потокового видео. В самом деле, если бы изображения в любой последовательности рассматривались просто как наборы отдельных изображений или «кадров», то эту новую задачу можно было бы уже считать решенной; поэтому особое значение приобрела интерпретация последовательности изображений как самостоятельного объекта. Возникла потребность в алгоритмах для идентификации и отслеживания движущихся объектов в любой последовательности изображений.

Мы могли бы решить эту задачу, идентифицируя объекты во всех кадрах, а затем вычисляя треки, показывающие, как объекты перемещались между кадрами. Этот подход может быть реализован в соответствии со следующим алгоритмом:

для всех кадров подряд
 найти и идентифицировать все объекты
 связать объекты между кадрами,
 перечислить все объекты и их треки.

Эта процедура требует, чтобы все объекты были обнаружены и распознаны, и в этом случае их связывание для формирования треков будет включать связывание только объектов одного и того же класса распознавания (например, автомобилей). Однако было бы еще лучше, если бы все объекты одного класса можно было идентифицировать по отдельности (например, автомобиль 1, автомобиль 2 и т. д.), хотя, если некоторые объекты очень похожи между собой, может возникнуть некоторая путаница при их связывании.

Эту довольно сложную процедуру можно было бы упростить, если бы объекты характеризовались параметрами их движения. На самом деле информация об отслеживании должна обеспечивать значительную экономию вычислений. Таким образом, мы приходим к альтернативной стратегии:

обнаружить все объекты в первом кадре
 найти, как эти объекты двигались в каждом последующем кадре
 перечислить все объекты и их треки.

Эта упрощенная процедура требует, чтобы объекты были *обнаружены* (а не распознаны) в первом кадре. Кроме того, нет необходимости повторять их распознавание в последующих кадрах, поскольку они должны однозначно идентифицироваться по их относительной близости.

Дальнейшая экономия усилий в принципе может быть достигнута за счет исключения первого этапа – обнаружения всех объектов в первом кадре. Все, что нам нужно сделать, – это обнаружить сами движения и идентифицировать все, что движется, как объект. Возможно, самый простой способ достичь этого – выделить различия между соседними кадрами, и в этом случае любые изменения должны указывать на расположение движущихся объектов. Однако этот подход имеет тенденцию обнаруживать только ограниченные участки контуров цели: например, он будет игнорировать большую часть любого объекта однородной интенсивности – в соответствии с известной формулой разности $\Delta I.v$. Самый простой выход из этой затруднительной ситуации – смоделировать фон; затем, вычитая каждый кадр из фоновой модели, мы сможем найти любые движущиеся объекты. (Естественно, это будет работать лучше всего, если фон останется неподвижным.)

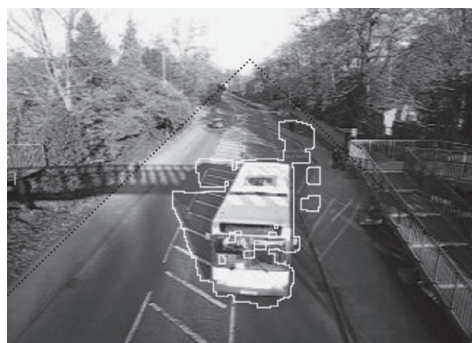
Хотя эта идея привлекательна, ее не так-то просто применить на практике. Одна из основных проблем заключается в том, что сцена, содержащая ряд движущихся объектов, сопровождается постоянным изменением модели фона, по мере того как по ней проходят движущиеся объекты: характерным примером является сцена дороги, по которой движутся транспортные средства. В этом случае незначительные изменения в выведенном фоне должны быть устранены каким-либо процессом усреднения. Обратите внимание, что полученный фон также необходимо будет обновлять с течением времени из-за изменений окружающего освещения и различных погодных условий. Отсюда вытекает вопрос, как реализовать это обновление, заодно компенсируя прошедшие объекты. Усреднение кадров во времени – плохой способ, но

некоторые исследования (Lo, Velastin, 2001; Cucchiara et al., 2003) показали, что временная *медианная* фильтрация может быть весьма эффективной, поскольку она способна устранять влияние выпадающих значений интенсивности, например из-за транспортных средств, движущихся на неподвижном фоне.

Одна из проблем с этим подходом заключается в том, что нахождение временной медианы требует хранения большого количества кадров – стратегия, которая не применима к обычному усреднению, которое удобно выполняется при помощи скользящего среднего, взвешенного по времени. Однако Дэвис (Davies, 2017) показал, что эту проблему можно решить путем итеративной реализации временной медианы и что это решение также позволяет обновить модель фона, чтобы компенсировать эффекты переменного фонового освещения. Кроме того, он обнаружил, что еще лучшим подходом является использование «сдержанного» медианного фильтра, в котором игнорируются экстремальные интенсивности и цвета в ограниченной полосе по сравнению с предыдущим итеративным приближением. В дорожной сцене это предотвращает чрезмерное искажение выводимых уровней фона транспортными средствами, которые временно неподвижны (см. рис. 1.27 и 1.28).

Заметим, что при таком подходе нет априорной причины, по которой интенсивность транспортного средства или другого объекта должна быть больше или меньше истинного фона – очевидно, что существуют обе возможности. Следовательно, вычитание текущего кадра из фоновой модели может разбить любой движущийся объект на несколько частей, которые впоследствии придется рекомбинировать с помощью таких методов, как морфологическая обработка. К счастью, этот метод также помогает устранить шум. Например, как видно по рис. 1.27 и 1.28, могут быть очень успешно подавлены эффекты движения листьев или ветвей на ветру.

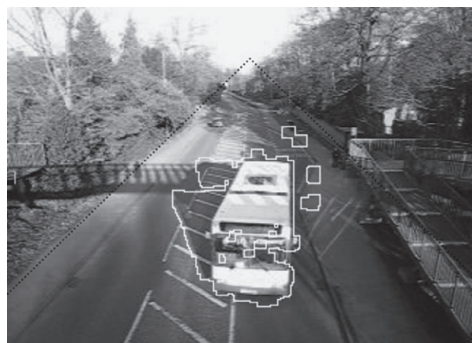
Интересно, что в то время как описанный выше подход может успешно узнавать автомобили в дорожных сценах, он рассматривает их тени как части объектов, поскольку не использует рассуждения более высокого уровня. Этот эффект проиллюстрирован на рис. 1.27 и 1.28. На самом деле обнаружение теней широко изучалось, и Хорпрасерт (Horprasert et al., 1999) продемонстрировал полезный принцип для его реализации: он основан на том факте, что тени имеют аналогичную цветность, но более низкую яркость, чем модель фона.



(a)



(b)



(c)



(d)

Рис. 1.27 ❖ Вычитание фона с использованием временного медианного фильтра. Обратите внимание на множество стационарных теней, которые полностью игнорируются в процессе вычитания фона. В (a) «призрак» автобуса (из его прежнего положения) все еще появляется, но в (b) он начал снова сливаться с фоном; задача значительно облегчается в (c) и (d), где применяется «сдержанный» временной медианный фильтр. В целом наихудшими проблемами являются фрагментация объектов переднего плана и ложные формы (включая эффекты движущихся теней). Линии черных пунктирных точек обозначают соответствующий участок дороги: почти вся качающаяся на ветру растительность находится за пределами этого участка

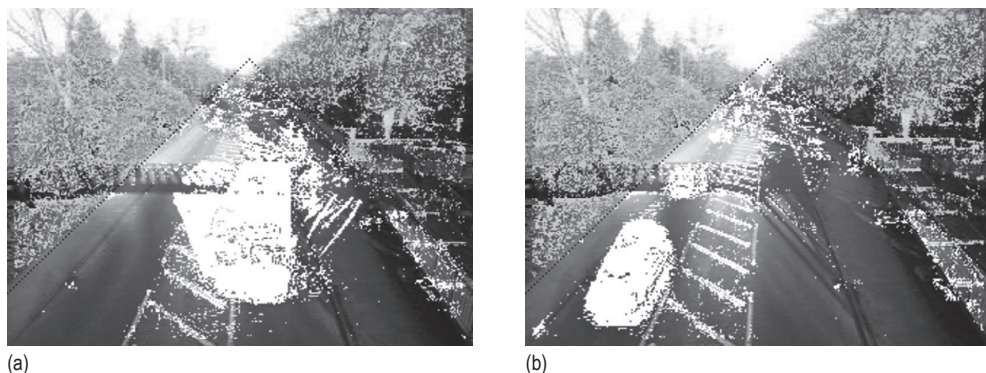


Рис. 1.28 ❖ Кадры (а) и (б) иллюстрируют трудности интерпретации непосредственных результатов вычитания фона. Эти два кадра ясно показывают проблемы с шумом, которые возникают во время вычитания фона: белые пиксели указывают, где текущий кадр не соответствует модели фона. Чтобы в значительной степени устранить шум и максимально интегрировать формы транспортного средства, используются морфологические операции (размытие с последующей дилатацией), как показано белыми графическими контурами на рис. 1.27. Обратите внимание, что последние содержат не только формы транспортных средств, но и тени, которые движутся вместе с ними

1.5.2. Альтернативы вычитанию фона

В то время как метод вычитания фона, описанный выше, по своей сути прост, удивительно эффективен и очень быстро работает, он также ограничен (а) предположением, что фон в основном не меняется, хотя и справляется (посредством усреднения во времени и морфологической обработки) с движущимися объектами фона, и (б) тем, что не использует надлежащие модели объектов переднего плана. Более строгий подход состоял бы в том, чтобы рассматривать распределения интенсивностей и цветов для любого пикселя как суперпозицию нескольких распределений, соответствующих двум или трем компонентным источникам. Здесь важно то, что каждое из распределений компонентов может быть достаточно узким и четко определенным. Это означает, что если каждый из них известен из продолжающегося обучения, можно проверить любую текущую интенсивность I , чтобы определить, соответствует ли она фону или новому объекту переднего плана.

Это делает *смешанные модели Гаусса* (Gaussian mixture models, GMM) полезными для представления истинных диапазонов интенсивности фона и переднего плана. На самом деле количество компонентов в любом пикселе изначально неизвестно: действительно, большая часть пикселей будет иметь только один компонент, но количество компонентов, требуемое на практике, обычно находится в диапазоне от 3 до 5. Однако определение GMM требует применения алгоритма *максимизации ожидания* (expectation maximization, EM) и является обременительным с вычислительной точки зрения. На самом деле, хотя для *инициализации* процесса генерации фона обычно используется этот строгий подход, многие процессы-воркеры используют более простые

и эффективные методы для его обновления, чтобы текущий процесс мог продолжаться в режиме реального времени.

К сожалению, подход GMM терпит неудачу, когда фон имеет очень высокие частотные вариации. По сути, это связано с тем, что алгоритм должен справляться с быстро меняющимися распределениями, которые могут сильно меняться за очень короткие промежутки времени, поэтому статистика становится слишком плохо определенной. Чтобы решить эту проблему, некоторые исследователи (Elgammal et al., 2000) отошли от параметрического подхода GMM. Их непараметрический метод включает в себя использование функции сглаживания ядра (обычно гауссовой) и для каждого пикселя применение ее к N выборкам из I для кадров, появляющихся в течение периода Δt до текущего момента времени t . Этот подход способен быстро адаптироваться к скачкам от одного значения интенсивности к другому, в то же время получая локальные отклонения для каждого пикселя. Таким образом, его ценность заключается в его способности забывать старые интенсивности и отражать локальные вариации, а не случайные скачки интенсивности. Кроме того, поскольку он не использует алгоритм EM, то может работать с высокой эффективностью в режиме реального времени и обеспечивать точное обнаружение объектов на переднем плане в сочетании с низким уровнем ложных срабатываний. Чтобы достичь таких показателей, он использует сначала отдельные функции гауссова ядра для каждого цветового канала, а затем упомянутый ранее метод на основе цветности для подавления теней.

До сих пор мы видели, что преимущества вычитания фона заключаются в простоте применения, скорости выполнения и высокой эффективности – вплоть до способности (с соответствующими алгоритмическими корректировками) справляться с медленно меняющимся фоновым освещением; устранять тени на заднем и переднем планах; подавлять «призраки», возникающие от временно остановившихся транспортных средств, а также эффекты развивающейся растительности. С другой стороны, он полагается на статичный фон, что, в свою очередь, означает использование только фиксированных камер. Кроме того, нет никакой гарантии, что все части объектов переднего плана будут иметь разную интенсивность по сравнению с фоном – фактор, который может вызвать разбиение объектов на фрагменты и приводит к необходимости морфологической обработки, которая сама по себе является достаточно специальным решением.

Еще одна проблема, возникающая при отслеживании на основе вычитания фона, заключается в том, что объекты переднего плана могут быть частично или полностью перекрыты другими объектами: это происходит, в частности, когда пешеходы перемещаются в людных местах. В лучшем случае это может привести к фрагментации треков, а в худшем – к неправильному соединению фрагментов. Заметим также, что движение некоторых объектов может вообще прекратиться. Ясно, что для решения этих задач нужны тщательно продуманные методы. Традиционный способ работы с оборванными треками заключался в использовании прогнозирующих фильтров, таких как фильтр Калмана, но они имеют ограниченное применение, поскольку в основном нацелены на оценку вероятностей соединения пар дорожек на основе единичных унимодальных гауссовых плотностей.

В целом эти критические замечания и проблемы требуют, чтобы временные различия были подкреплены сопоставлением корреляции шаблонов для уверенности, что мы имеем дело с одним и тем же объектом. Как отмечают Липтон и соавторы (Lipton et al., 1998), для обнаружения движущихся объектов может использоваться временная разность (или родственные методы, такие как вычитание фона), а корреляционное сопоставление может использоваться (а) для точного определения местоположения объектов и (б) для обучения шаблона корреляции; на каждом этапе используется шаблон с наилучшей корреляцией как для (а), так и для (б). Фактически использование лучшего шаблона корреляции применимо, даже когда один объект частично перекрывает другой. Кроме того, это особенно полезно, когда конкретный объект становится неподвижным, поскольку тогда нет неопределенности в отношении его идентичности или местоположения.

Сталдер и др. (Stalder et al., 2009) разработали альтернативную стратегию отслеживания объектов, которую они описали как «отслеживание путем обнаружения». Цель состояла в том, чтобы позволить трекеру адаптироваться к любым изменениям внешнего вида объекта путем обновления модели объекта. Однако это приводит к так называемой «проблеме обновления шаблона», когда существует компромисс между адаптивностью и стабильностью (в частности, трекер может получить полностью искаженную, нежизнеспособную версию профиля объекта – процесс, называемый «дрейфом»). Эта трудность преодолевается путем переформулирования отслеживания как задачи частично контролируемого обучения, в которой во время отслеживания могут использоваться как размеченные, так и неразмеченные данные. Чтобы сделать эту работу, используется частичный бустинг¹ модели, при этом каждому неразмеченному образцу в области локального поиска назначается псевдометка y_i и вес важности λ_i (меченые образцы имеют метку y_i и важность 1). После начального обнаружения, приводящего к предварительному H_p , строится классификатор объектов H с учетом положительных выборок от объекта и отрицательных от окружающего фона: обычно для указания нового положения объекта берется локальный максимум доверительного распределения, и класс обновляется. Таким образом, мы получаем последовательность местоположений объекта, начиная с первого (получившегося в результате первоначального обнаружения), а затем переходим ко многим последующим отслеживаемым позициям. Однако, в принципе, уравнения бустинга также могут привести к тому, что первичное обнаружение (приор) либо исчезнет, либо будет слишком сильно доминировать, что соответствует, соответственно, дрейфу или нулевой адаптации. Но в целом дрейф ограничен, так как трекер не может уйти слишком далеко от приора.

Проблемы с описанной выше стратегией отслеживания включают (а) принятие частичных окклюзий или явно допустимых изменений внешнего вида за недопустимый дрейф и (б) перескакивание на похожие объекты (например, детектор лиц может перескакивать с лица одного человека на другое).

¹ Бустинг – композиционный метаалгоритм машинного обучения, применяется главным образом для уменьшения смещения, а также дисперсии в обучении с учителем. – Прим. перев.

Очевидно, что отслеживание путем обнаружения в основном применимо для отслеживания одной цели, но когда требуется сопровождение нескольких целей, необходимо учитывать все три процесса: обнаружение, отслеживание и распознавание. В частности, узнавание следует понимать как различение сходных объектов в сцене.

Система классификации множественных целей Сталдера и др. (Stalder et al., 2009) продемонстрировала очень хорошую работоспособность при отслеживании множества объектов. Она ограничивала дрейф за счет тщательного использования контролируемых обновлений и избегания петель обратной связи, тем самым предотвращая накопление небольших ошибок, которые могли привести к дрейфу. Это было достигнуто за счет того, что трекер стал доминирующим элементом в подходе, так что его информационный поток возвращался (в цикле) обратно к себе или (в конечном итоге) к предыдущему распознавателю. Эта система позволяла отслеживать объекты с помощью движущейся камеры, а также была достаточно надежной, чтобы обеспечить долгосрочное отслеживание в течение 24 часов.

Еще одним аспектом сопровождения множественных целей является возможность повторной идентификации объектов, которые временно скрыты от поля зрения (например, временно заслонены или оказались вне поля зрения камеры). Решение включает сопоставление идентификации и считается допустимым только в том случае, если достигнутая степень совпадения значительно выше, чем для любых других потенциальных совпадений. Повторная идентификация может завершиться ошибкой, если степень совпадения станет слишком разной во время соответствующего временного промежутка. Опять же, система Стадлера также показала хорошие результаты в этом отношении.

Калал и соавторы (Kalal et al., 2011) разработали мощную систему отслеживания, предназначенную, в частности, для устранения ошибок дрейфа во время выполнения и проблем, возникающих, когда отслеживаемые объекты исчезают из поля зрения. Их подход был аккуратно описан как «отслеживание-обучение-обнаружение». Ключевым аспектом этого метода является то, что обучение осуществляется «Р-экспертом», который оценивает пропущенные обнаружения, «N-экспертом», который оценивает ложные тревоги, и средствами обновления обоих экспертов путем обучения. Разделяя отслеживание и обнаружение, они утверждали, что ни возможности отслеживания, ни возможности обнаружения не снижены, и представили убедительные доказательства успеха их подхода, в частности оценив точность многих наборов данных в среднем на уровне 81 %, тем самым значительно превзойдя пять более ранних подходов, ни один из которых не достигал точности выше 22 %.

В то время как в работе Калала отслеживали только отдельные объекты, другие исследователи (Wu et al., 2012) показали, как выполнять обнаружение связи и ассоциации данных для нескольких объектов, чтобы обеспечить определение полных треков. Этот подход использовал ассоциацию данных о сетевых потоках и опирался на более раннюю статью Кастаньона (Castañón, 1990), озаглавленную «Эффективный алгоритм поиска k лучших путей через решетку». Это название говорит о том, что метод в значительной степени

основан на обширном анализе графов: хотя это интересно, объем книги не позволяет включить здесь полное обсуждение данных методов. Достаточно сказать, что отслеживание нескольких объектов – сложная задача, которая становится еще более сложной по мере увеличения количества целей и заполнения всей сцены объектами и треками. Таким образом, в этом отношении работу Ву можно считать более тщательной, чем работу Стадлера. Мы изучим более свежие разработки по этой теме в части F, раздел 1.7.7, после рассмотрения методов глубокого обучения.

1.6. Часть Е. АНАЛИЗ ТЕКСТУР

1.6.1. Введение

Мы уже рассмотрели несколько основных аспектов анализа изображений, включая, в частности, обнаружение признаков, распознавание объектов и сегментацию. Теперь переходим к анализу текстур. Мы начнем с определения текстуры как характерного изменения интенсивности, которое должно позволить нам распознать и описать текстурированную область и очертить ее границы (рис. 1.29). Как правило, текстура представляет собой образец интенсивности, возникающий при отражении света от поверхности, имеющей определенную степень шероховатости. Ясно, что гладкая, равномерно освещенная поверхность не будет иметь текстуры, в то время как кусок ткани или песчаный пляж будут иметь собственные характерные текстуры.

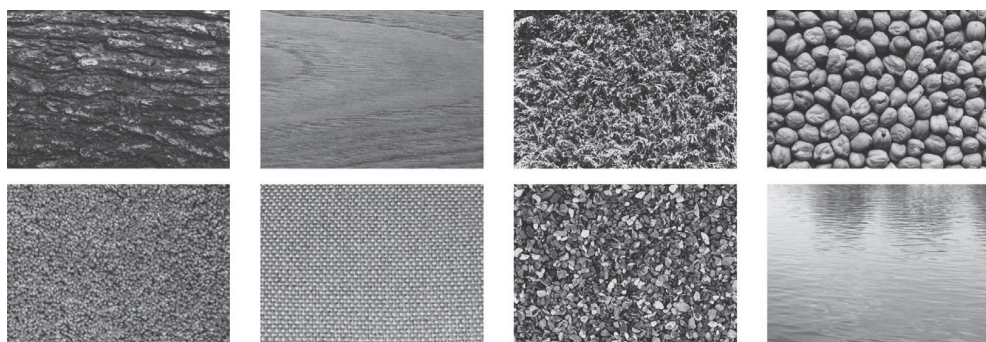


Рис. 1.29 ❖ Разнообразие текстур. На этих изображениях представлены различные знакомые текстуры, которые легко распознаются по их характерным образцам интенсивности

Вообще говоря, текстуры различаются по степени случайности и регулярности, и в последнем случае они могут иметь высокую или низкую направленность. Например, куски ткани обычно демонстрируют высокую степень регулярности и направленности, в то время как картина интенсивности, исходящая от поверхности песчаного пляжа, может казаться в высшей степени

случайной с незначительной направленностью. Другим фактором является масштаб воспринимаемого размера фрагментов для поверхности, который будет небольшим для песка и намного больше для лотка с горохом. Крошечные элементы, составляющие текстурированную поверхность, часто называют текстурными элементами, или *текселями* (texel). Из вышеупомянутых соображений можно вывести следующие характеристики текстур:

- 1) тексели будут иметь различные размеры и степени однородности;
- 2) тексели будут ориентированы в разных направлениях;
- 3) тексели будут располагаться на разном расстоянии в разных направлениях;
- 4) контраст будет иметь различные величины и вариации;
- 5) между текселями может проглядывать разное количество фона;
- 6) вариации, составляющие текстуру, могут иметь разную степень регулярности либо случайности.

Значительное количество параметров, необходимых для описания текстуры, неизбежно делает анализ текстур довольно сложным; и, конечно, многие параметры будут иметь высокую степень изменчивости, поэтому анализ часто приводит к статистическому описанию текстур. В следующем разделе показаны некоторые способы решения этой задачи.

1.6.2. Основные подходы к анализу текстур

В разделе 1.6.1 мы определили текстуру как характерное изменение интенсивности области изображения, которое должно позволить нам распознать ее, описать и очертить ее границы. Ввиду статистической природы текстур это побуждает нас характеризовать текстуру дисперсией значений интенсивности, взятых в области текстуры. Однако такой подход не даст достаточно богатого описания текстуры для большинства целей: он также будет непригоден в тех случаях, когда тексели хорошо определены или когда в текстуре присутствует высокая степень периодичности. С другой стороны, для очень периодических текстур, таких как многие ткани, естественно рассмотреть возможность использования анализа Фурье. К сожалению, хотя этот подход давно и тщательно протестирован, результаты не были обнадеживающими.

Автокорреляция – еще один очевидный подход к анализу текстуры, поскольку он должен выявлять как локальные вариации интенсивности, так и повторяемость текстуры (рис. 1.30). Одно из первых исследований было проведено Кайзером (Kaizer, 1955). Он исследовал, на сколько пикселей должно сместиться изображение, прежде чем автокорреляционная функция упадет до $1/e$ от своего начального значения, и предложил основанную на этом субъективную меру *грубости*. Однако Розенфельд и Трой (1970) показали, что автокорреляция не является удовлетворительной мерой грубости. Кроме того, автокорреляция не является хорошим дискриминатором изотропии в естественных текстурах. Поэтому исследователи быстро переняли матричный подход, предложенный Хараликом (Haralick et al., 1973). Подход с использованием матрицы совпадений на уровне серого основан на изучении статистики распределения интенсивности пикселей. Как упоминалось

выше, статистика отдельных пикселей не обеспечивает достаточно подробного описания текстур для практических приложений. Таким образом, естественно рассматривать статистику второго порядка, полученную при рассмотрении *пар* пикселей с определенными пространственными отношениями друг к другу. Поэтому в подходе Харалика используются *матрицы совпадений*, которые выражают относительные частоты $P(i, j | d, \theta)$, с которыми два пикселя, имеющих относительные полярные координаты (d, θ) , появляются с интенсивностью i, j . Матрицы совпадений предоставляют необработанные числовые данные о текстуре, хотя эти данные должны быть сжаты до относительно небольшого количества числовых значений, прежде чем их можно будет использовать для классификации текстуры. В работе Харалика описано четырнадцать таких мер, и они успешно использовались для классификации многих типов материалов (включая, например, древесину, кукурузу, траву и воду).

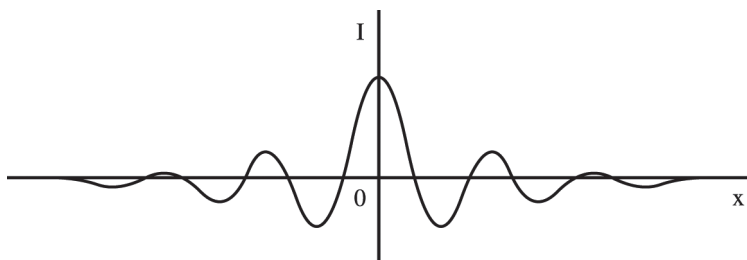


Рис. 1.30 ❖ Использование функции автокорреляции для анализа текстуры. На этой диаграмме показан возможный одномерный профиль автокорреляционной функции для фрагмента ткани, в которой переплетение подвержено значительным пространственным вариациям: обратите внимание, что периодичность автокорреляционной функции затухает на довольно коротком расстоянии

К сожалению, количество данных в матрицах совпадений может быть во много раз больше, чем в исходном изображении, – ситуация, которая усугубляется в более сложных случаях количеством значений d и θ , необходимых для точного представления текстуры. Кроме того, число уровней серого обычно равно 256, а количество матричных данных зависит от квадрата этого числа. Наконец, матрицы совпадения просто обеспечивают новое представление: сами по себе они не решают проблему распознавания. В силу этих причин в 1980-е гг. появилось весьма значительное разнообразие методов анализа текстур. Среди них метод Лоуза (Laws, 1979; 1980a; 1980b) выделяется тем, что он привел к другим разработкам, обеспечивающим систематические, адаптивные средства анализа текстуры. Этот подход рассматривается в следующем разделе.

1.6.3. Метод Лоуза на основе энергии текстуры

В 1979 и 1980 гг. Лоуз представил свой новый подход к текстурному анализу, основанный на так называемой *энергии текстуры* (Laws, 1979, 1980a,b). Подход включал применение простых фильтров к цифровым изображениям. Основные фильтры, которые он использовал, были обычными фильтрами Гаусса, детектором краев и фильтрами по типу Лапласа, специально разработанными для выделения точек с высокой «текстурной энергией» на изображении. Выявив эти точки с высокой энергией, сгладив различные отфильтрованные изображения и объединив полученную информацию, Лоуз смог очень эффективно охарактеризовать текстуры. Как отмечалось ранее, подход Лоуза оказал большое влияние на многие последующие работы, и поэтому стоит рассмотреть его здесь более подробно.

Маски Лоуза создаются путем свертывания всего трех основных масок 1×3 :

$$L3 = [1 \ 2 \ 1]; \quad (1.104)$$

$$E3 = [-1 \ 0 \ 1]; \quad (1.105)$$

$$S3 = [-1 \ 2 \ -1]. \quad (1.106)$$

Начальные буквы этих масок обозначают локальное усреднение (Local), обнаружение краев (Edge detection) и обнаружение пятна (Spot detection). Фактически эти базовые маски охватывают все подпространство 1×3 и образуют полный набор. Точно так же маски 1×5 , полученные путем свертки пар этих масок 1×3 , вместе образуют полный набор, в котором только следующие пять различны:

$$L5 = [1 \ 4 \ 6 \ 4 \ 1]; \quad (1.107)$$

$$E5 = [-1 \ -2 \ 0 \ 2 \ 1]; \quad (1.108)$$

$$S5 = [-1 \ 0 \ 2 \ 0 \ -1]; \quad (1.109)$$

$$R5 = [1 \ -4 \ 6 \ -4 \ 1]; \quad (1.110)$$

$$W5 = [-1 \ 2 \ 0 \ -2 \ 1]. \quad (1.111)$$

(Здесь начальные буквы такие же, как и раньше, с добавлением обнаружения пульсаций (Ripple detection) и обнаружения волн (Wave detection).) Мы также можем использовать матричное умножение, чтобы объединить маски 1×3 и аналогичный набор масок 3×1 и получить девять масок 3×3 , например:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}. \quad (1.112)$$

Результирующий набор масок снова образует полный набор (табл. 1.2). Обратите внимание, что две из этих масок идентичны маскам оператора Собеля. Соответствующие маски 5×5 полностью аналогичны, но здесь подробно не рассматриваются, поскольку все необходимые принципы охватываются масками 3×3 .

Таблица 1.2. Девять масок Лоуза 3×3

L3^TL3	L3^TE3	L3^TS3
1 2 1	-1 0 1	-1 2 -1
2 4 2	-2 0 2	-2 4 -2
1 2 1	-1 0 1	-1 2 -1
E3^TL3	E3^TE3	E3^TS3
-1 -2 -1	1 0 -1	1 -2 1
0 0 0	0 0 0	0 0 0
1 2 1	-1 0 1	-1 2 -1
S3^TL3	S3^TE3	S3^TS3
-1 -2 -1	1 0 -1	1 -2 1
2 4 2	-2 0 2	-2 4 -2
-1 -2 -1	1 0 -1	1 -2 1

Все подобные наборы масок включают одну, компоненты которой не усредняются до нуля. Этот метод менее полезен для анализа текстуры, поскольку он дает результаты, зависящие больше от интенсивности изображения, чем от текстуры. Остальная часть результата чувствительна к краевым точкам, пятнам, линиям и их комбинациям.

После создания изображений, указывающих на локальную резкость и т. д., следующим этапом является вывод локальных величин этих параметров. Затем эти величины сглаживаются по области, которая больше размера маски основного фильтра (например, Лоуз использовал окно сглаживания 15×15 после применения своих масок 3×3): в результате этого сглаживаются промежутки между краями текстуры и другие микропризнаки. К этому моменту исходное изображение преобразовано в векторное изображение, каждый компонент которого представляет «энергию» определенного типа. В то время как Лоуз (1980b) использовал для оценки энергии текстуры как квадраты величин, так и абсолютные величины, первые соответствуют истинной энергии и дают лучший отклик, вторые полезны тем, что требуют меньше вычислений:

$$E(l, m) = \sum_{i=l-p}^{l+p} \sum_{j=m-p}^{m+p} |F(i, j)|, \quad (1.113)$$

где $F(i, j)$ – локальная величина типичного микропризнака, которая сглаживается в общей позиции сканирования (l, m) в окне $(2p + 1) \times (2p + 1)$.

На следующем этапе требуется комбинировать различные энергии разными способами, предоставляя несколько выходных данных, которые могут

быть переданы в классификатор для принятия решения о конкретном типе текстуры в каждом местоположении пикселя (рис. 1.31): при необходимости используется анализ главных компонент, чтобы помочь выбрать подходящий набор промежуточных результатов.

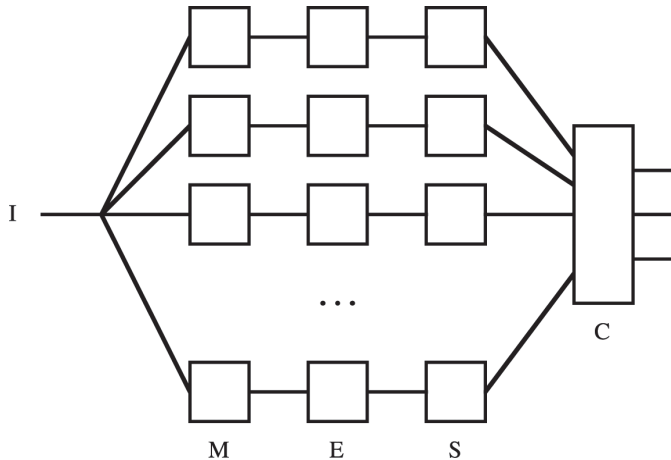


Рис. 1.31 ❖ Базовая форма классификатора текстур Лоуза. Здесь I – входящее изображение, M – расчет микропризнаков, E – расчет энергии, S – сглаживание, C – окончательная классификация

Метод Лоуза обеспечил превосходную точность классификации на уровне 87 % по сравнению с 72 % для метода матрицы совпадения применительно к составному текстурному изображению травы, рафии, песка, шерсти, свиной кожи, воды и древесины (Laws, 1980). Лоуз также обнаружил, что выравнивание гистограммы, обычно применяемое к изображениям для устранения различий первого порядка в распределении оттенков серого поля текстуры, привело к небольшому улучшению. В независимом исследовании (Pietikäinen et al., 1983) было подтверждено, что измерения энергии текстуры Лоуза более эффективны, чем измерения, основанные на парах пикселей (в частности, матрицы совпадения).

1.6.4. Метод собственного фильтра Аде

В 1983 году Аде исследовал теорию, лежащую в основе метода Лоуза, и разработал пересмотренное обоснование с точки зрения *собственных фильтров* (eigenfilter). Он взял все возможные пары пикселей в окне 3×3 и охарактеризовал данные интенсивности изображения ковариационной матрицей 9×9 . Затем определил *собственные векторы* (eigenvector), необходимые для диагонализации этой матрицы. Они соответствуют маскам фильтров, аналогичным маскам Лоуза, т. е. использование этих масок собственных фильтров создает изображения, которые являются изображениями главных компонент для данной текстуры. Кроме того, каждое *собственное значение* (eigenvalue)

дает ту часть дисперсии исходного изображения, которая может быть извлечена соответствующим фильтром. По сути, дисперсии дают исчерпывающее описание данной текстуры с точки зрения текстуры изображений, из которых первоначально была получена ковариационная матрица. Ясно, что фильтры, обеспечивающие низкую дисперсию, можно считать относительно неважными для распознавания текстур.

Будет полезно проиллюстрировать технику для окна 3×3 . Здесь мы следуем Аде (Ade, 1983) в нумерации пикселей в окне 3×3 в порядке сканирования:

1	2	3
4	5	6
7	8	9

Это приводит нас к ковариационной матрице 9×9 для описания взаимосвязей между интенсивностями пикселей в окне 3×3 , как указано выше. Здесь мы вспоминаем, что описываем текстуру, и, предполагая, что ее свойства не синхронны с *тесселяцией*¹ пикселей, мы ожидаем, что различные коэффициенты ковариационной матрицы **C** будут равны. На самом деле существует только 12 различных пространственных отношений между пикселями, если мы не принимаем во внимание перемещения целых пар, или 13, если мы включим в набор нулевой вектор (табл. 1.3). Таким образом, ковариационная матрица, компоненты которой включают 13 параметров $a-m$, принимает вид:

$$C = \begin{bmatrix} a & b & f & c & d & k & g & m & h \\ b & a & b & e & c & d & l & g & m \\ f & b & a & j & e & c & i & l & g \\ c & e & j & a & b & f & c & d & k \\ d & c & e & b & a & b & e & c & d \\ k & d & c & f & b & a & j & e & c \\ g & l & i & c & e & j & a & b & f \\ m & g & l & d & c & e & b & a & b \\ h & m & g & k & d & c & f & b & a \end{bmatrix}. \quad (1.114)$$

Таблица 1.3. Пространственные отношения между пикселями в окне 3×3

a	b	c	d	e	f	g	h	i	j	k	l	m
9	6	6	4	4	3	3	1	1	2	2	2	2

В табл. 1.4 показано количество вхождений пространственных отношений между пикселями в окне 3×3 . Обратите внимание, что a – это диагональный элемент ковариационной матрицы **C**, а все остальные элементы встречаются в **C** в два раза чаще, чем указано в таблице.

¹ Тесселяция – мозаичное заполнение, разбиение плоскости картины на фрагменты, заполняющие картину без каких-либо наложений или пробелов. – Прим. перев.

Матрица C симметрична; собственные значения действительной симметричной ковариационной матрицы действительны и положительны, а собственные векторы взаимно ортогональны. Кроме того, полученные таким образом собственные фильтры отражают правильную структуру изучаемой текстуры и идеально подходят для ее охарактеризования. Например, для текстуры с ярко выраженным высоконаправленным узором будет одно или несколько собственных значений высокой энергии с собственными фильтрами, имеющими сильную направленность в соответствующем направлении.

1.6.5. Сравнение методов Лоуза и Аде

На этом этапе полезно более тщательно сравнить подходы Лоуза и Аде. В методе Лоуза используются стандартные фильтры, создаются изображения энергии текстуры, а *затем* может применяться метод главных компонент с целью распознавания; в свою очередь, в методе Аде применяются специальные фильтры (собственные фильтры), включающие результаты применения метода главных компонент, после чего вычисляются меры энергии текстуры, и подходящее их количество применяется для распознавания.

Подход Аде превосходит тем, что позволяет на раннем этапе исключить малозначащие компоненты, тем самым экономя вычисления. Например, в приложении Аде первые пять из девяти компонентов содержат 99,1 % всей энергии текстуры, поэтому на остальные можно не обращать внимания; кроме того, оказалось, что еще две компоненты, содержащие соответственно 1,9 % и 0,7 % энергии, также можно было бы игнорировать с небольшой потерей точности распознавания. Однако в некоторых приложениях текстуры могут постоянно меняться, и может оказаться нецелесообразным точно настраивать метод для конкретных данных, относящихся к любому моменту времени. (Например, эти замечания относятся (1) к тканям, степень растяжения которых постоянно меняется в процессе производства, (2) к сырым пищевым продуктам, таким как бобы, размер которых зависит от источника поставки, и (3) к переработанным пищевым продуктам, таким как пирожные, рассыпчатость которых зависит от температуры приготовления и содержания водяного пара.)

Унзер (Unser, 1986) разработал более общий вариант техники Аде. В этом подходе производительность оптимизирована не только для классификации текстур, но и для распознавания двух текстур путем одновременной диагонализации двух ковариационных матриц. Этот метод получил дальнейшее развитие в следующих работах (Unser, Eden 1989; 1990), в которых проводится тщательный анализ использования нелинейных детекторов. В результате авторы пришли к использованию двух уровней нелинейности: один применяется сразу после линейных фильтров и разработан (путем использования специальной гауссовой текстурной модели) для подачи на этап сглаживания подлинной дисперсии или других подходящих показателей, а другой – после этапа пространственного сглаживания для компенсации эффекта предыдущего фильтра в стремлении обеспечить значение функции в тех же единицах, что и входной сигнал. С практической точки зрения это означает

возможность получения среднеквадратичного сигнала текстуры от каждого из каналов линейного фильтра.

В целом метод Лоуза возник в 1980-х гг. как серьезная альтернатива методу матриц совместной встречаемости. Также следует отметить, что еще были разработаны альтернативные многообещающие методы, например метод принудительного выбора Вистнеса (Vistnes, 1989) для нахождения краев между различными текстурами, который, по-видимому, имеет значительно лучшую точность, чем метод Лоуза. В своем исследовании Вистнес делает вывод о том, что метод Лоуза ограничен (а) малым масштабом масок, которые могут пропускать более крупномасштабные текстурные фрагменты, и (б) тем фактом, что операция сглаживания энергии текстуры размывает значения признаков текстуры по всему краю. Последний вывод (или даже худшая ситуация, когда третий класс текстур оказывается расположенным в области границы между двумя текстурами) также был отмечен Сяо и Савчуком (Hsiao, Sawchuk, 1989), которые применили усовершенствованный метод сглаживания признаков.

1.6.6. Последние разработки

В 2000-х гг. появилась тенденция к анализу текстур, не зависящих от масштаба и вращения. В частности, в статье Дженни и Джирса (Janney, Geers, 2010) описан подход «инвариантных признаков локальных текстур», использующий строго круговой одномерный массив точек выборки вокруг любой заданной позиции. Метод использует вейвлеты Хаара и, как результат, эффективен в вычислительном отношении. Он применяется в нескольких масштабах для достижения масштабной инвариантности; кроме того, выполняется нормализация интенсивности, чтобы сделать метод инвариантным по освещению. Также следует отметить книгу (Mirmehdi et al., 2008), посвященную этому довольно узкому вопросу. Она представляет собой редактируемый сборник работ, содержащий вклад различных исследователей и обобщающий положение дел до 2010 г. Рассмотрение более поздних достижений по этой теме мы отложим до части F, разделов 1.7.8 и 1.7.9, после введения в методы глубокого обучения.

1.7. Часть F. От искусственных нейронных сетей к методам глубокого обучения

1.7.1. Введение: как ИНС превратились в СНС

Первоначальная цель разработки *искусственных нейронных сетей* (ИНС) состояла в том, чтобы имитировать процессы, происходящие в зрительной системе человека. На первый взгляд, зрительная система мозга устроена и работает настолько просто – целые сцены анализируются «с одного взгля-

да» без видимых усилий, – что возникает вполне естественное желание построить аналог зрительной системы на основе компьютера. Ясно, что ИНС, предназначенная для имитации зрительной системы человека, должна состоять из нескольких слоев, каждый из которых изменяет данные сначала локально, а затем все большими и большими наборами нейронов, пока не будут выполнены такие задачи, как распознавание и анализ сцены. Однако в первое время использование ИНС, как правило, ограничивалось очень небольшим количеством слоев: рабочая максимальная глубина состояла из одного входного слоя, трех скрытых слоев и одного выходного слоя, хотя позже было обнаружено, что многие основные задачи могут быть решены с использованием трехслойной сети с одним скрытым слоем.

Одной из причин ограничения количества слоев была *задача назначения коэффициентов доверия* (credit assignment problem), которая означала, что стало сложнее обучать слои «сквозь» несколько предшествующих слоев; в то же время большее количество слоев означало, что нужно обучить больше нейронов и для выполнения задачи требовалось больше вычислений. Поэтому ИНС, как правило, предназначались для выполнения только классического процесса распознавания и получали входные данные от детекторов признаков, применяемых на предыдущих уровнях, не связанных с обучением. Стандартной парадигмой был препроцессор изображений, за которым следовал обученный классификатор. Поскольку очень хорошие детекторы признаков можно было спроектировать вручную, это не вызывало никаких очевидных проблем. Однако с течением времени возникла потребность в полномасштабном анализе реальных сцен, которые могли бы содержать изображения многих типов объектов во многих положениях. Таким образом, нарастала потребность в переходе к гораздо более сложным многоуровневым системам распознавания, для которых ранние модификации ИНС были непригодны. Также появилась необходимость в обучении самой системы предварительной обработки, чтобы она точно соответствовала требованиям следующей системы анализа объектов; иными словами, возникла необходимость в создании интегрированных многослойных нейронных сетей.

Фактически к концу 1990-х гг. перспективы ИНС не вызывали оптимизма, потому что с ними конкурировали другие успешные методы, такие как *машины опорных векторов* (support vector machines, SVM). Кроме того, не был разработан научно обоснованный способ определения минимально необходимого количества слоев или нейронов и не было четкого понимания внутренних принципов работы ИНС. В результате специалисты, которые могли бы использовать их, не знали, насколько они надежны, и не были уверены, что смогут использовать их в реальных приложениях, поэтому ИНС начали терять популярность.

Важной причиной этого был еще и тот факт, что их архитектура и обучение давали плохую пространственную инвариантность для изображений. В частности, нейроны обучались индивидуально: каждый нейрон видел обучающие данные, отличные от других нейронов в своем слое; кроме того, веса связей между нейронами необходимо было инициализировать случайным образом. Эти факторы не позволяли получить одно и то же решение в отношении любого объекта независимо от его положения на изображении. Однако иссле-

дователи и разработчики нейросетей не стояли на месте, и в конце 2000-х гг. на первый план вышли сети с «глубоким» обучением (*глубокая нейронная сеть* (ГНС) – это сеть, в которой более трех нелинейных скрытых слоев, что выходит за рамки обычных ИНС).

Новым типом архитектуры стала *сверточная нейронная сеть* (convolutional neural network, CNN). Во многих отношениях это была менее требовательная архитектура, поскольку (а) каждый нейрон CNN не должен быть подключен ко *всем* выходам предыдущего слоя нейронов; (б) нейроны имеют одинаковые весовые параметры по всему слою. Тем не менее CNN по-прежнему используют обучение с учителем (supervised learning) и сеть обучается с помощью *обратного распространения ошибки* (backpropagation).

Важно отметить, что применение одинаковых весовых коэффициентов нейронов во всем слое значительно уменьшило общее количество параметров во всей сети и существенно упростило ее обучение; кроме того, может быть использовано большее количество слоев. Предоставление нейронам локальных связей еще больше улучшило ситуацию. Заметим, что если нейроны и веса идентичны во всем слое, результирующая математическая операция по определению является сверткой – отсюда и термин «сверточная нейронная сеть».

Еще одна особенность CNN заключается в том, что они используют функцию ReLU, а не сигмоидальные выходные функции. «ReLU» означает «Rectified Linear Unit» (спрямленный линейный блок) и определяется как $\max(0, x)$, где x – выходное значение непосредственно предшествующего слоя свертки. Эта функция ценна тем, что требует меньше вычислений, чем прежняя сигмовидная функция, и в то же время меньше искажает большие сигналы. По существу, функция ReLU позволяет избежать проблем с насыщением, которым подвержены ИНС (нейрон, дающий выходной сигнал, близкий к пределу (± 1) функции гиперболического тангенса, имеет тенденцию «застывать» на одном и том же значении, потому что нет градиента, чтобы увести алгоритм обратного распространения ошибки подальше от этой точки).

CNN также включают *пулинг* (pooling, объединение), т. е. получение всех выходных данных из локальности и получение из них одного выходного значения: обычно пулинг принимает форму суммы или операции нахождения максимума (max) над всеми входными данными, причем max-пулинг более распространен, чем операция суммирования или усреднения. Пулинг обычно выполняется в окнах 2×2 или 3×3 , первый вариант встречается чаще. Эти методы были направлены на минимальное изменение данных, чтобы удалить большую часть избыточности на определенном уровне сети, в то же время сохранив наиболее полезные данные.

Несколько сверточных слоев могут быть размещены сразу друг за другом, что делает их эквивалентными одной большей свертке – фактор, который может быть полезен для реализации более крупных признаков в одной и той же CNN. В целом CNN представляют собой разумную альтернативу ANN. Кроме того, они выглядят лучше приспособленными к идее постепенного перехода от локальных к глобальным операциям с изображениями и поиску все более и более крупных признаков или объектов в процессе.

Хотя продвижение по сети ведет нас от локальных операций к более глобальным, первым нескольким слоям CNN также свойственно искать опреде-

ленные низкоуровневые признаки; следовательно, они обычно имеют размеры, соответствующие размерам определенного типа изображения. Далее в сети обычно применяют операции пулинга, тем самым уменьшая размеры последующих слоев. После нескольких этапов свертки и пулинга сеть значительно сузится, поэтому можно сделать последние несколько слоев *полностью связанными*, т. е. такими, где в любом слое каждый нейрон подключен ко *всем* выходам предыдущего слоя. На этом этапе, скорее всего, будет относительно немного выходных данных, а те, которые останутся, будут определяться любыми параметрами, которые должны быть предоставлены сетью: они могут включать классификации и связанные с ними параметры, такие как абсолютные или относительные позиции.

1.7.2. Параметры, определяющие архитектуру CNN

При анализе архитектур CNN есть ряд моментов, заслуживающих внимания. В частности, необходимо определить несколько величин и терминов – *ширину* W , *высоту* H , *глубину* N , *страйд* (stride, шаг) S , *ширину паддинга* (заполнения нулями) P и *поле восприятия* R . Фактически ширина и высота – это просто размеры входного изображения либо размеры определенного слоя нейронной сети. Глубина N сети или конкретного блока в ней – это количество содержащихся в ней слоев.

Ширина W и высота H слоя – это количество нейронов в каждом измерении. Страйд S – это расстояние между соседними нейронами в выходном поле, измеренное в единицах, соответствующих расстоянию между соседними нейронами в поле ввода; страйд S можно определить по ширине и высоте, но обычно он одинаков для каждого измерения. Если $S = 1$, соседние слои имеют одинаковые размеры (но ниже мы покажем, как размер поля восприятия R может изменить это). Обратите внимание, что увеличение S может быть полезным, так как это экономит память и вычисления. В принципе, достигается эффект, аналогичный пулингу. Однако пулинг предполагает некоторое усреднение, а увеличение S просто уменьшает количество взятых выборов.

Параметр R_i – это ширина поля восприятия (или *рецептивного поля*) для каждого нейрона слоя i , т. е. количество входов для всех нейронов на этом уровне. Паддинг, или заполнение нулями, – это добавление P «виртуальных» нейронов, предоставляющих статические входные данные на каждом конце измерения ширины: им присваиваются фиксированные нулевые веса. Идея паддинга состоит в том, чтобы гарантировать, что все нейроны в одном слое имеют одинаковое количество входных данных, тем самым облегчая программирование. Однако это также гарантирует, что последовательные свертки не приведут к уменьшению активной ширины; в частности, когда $S = 1$, это позволяет нам сделать ширины соседних слоев в точности равными (т. е. $W_{i+1} = W_i$). Простая формула связывает несколько таких величин:

$$W_{i+1} = (W_i + 2P_i - R_i)/S_i + 1, \quad (1.115)$$

где разности относятся к входным данным слоя i и выходным данным слоя $i + 1$. Стоит подчеркнуть нулевую ситуацию $W_{i+1} = W_i$, которая возникает,

когда $S_i = 1$, $R_i = 1$ и $P_i = 0$. В общем случае цель паддинга состоит в том, чтобы учесть влияние крайних точек каждого слоя, при условии что количество нулей соответствует желаемым значениям страйда и размерности поля восприятия.

Наконец, необходимо сделать важное замечание об определении глубины слоев CNN. Предшествующее обсуждение подразумевало, что доступ к ряду смежных слоев CNN обычно осуществляется последовательно один за другим – как это действительно было бы в случае, если бы все более и более крупные свертки реализовывались одна за другой в попытке обнаружить все более и более крупные признаки или даже объекты. Однако есть и другая возможность: различные слои загружаются параллельно из заданной начальной точки в сети, например из входного изображения. Такая ситуация обычно возникает, когда изображение нужно искать по целому ряду различных признаков, таких как линии, края или углы, и результаты параллельно передаются на более обобщенный детектор. Эта стратегия была принята в архитектуре LeNet, которую Ян Лекун с соавторами (LeCun et al., 1998) разработал для распознавания рукописных цифр и почтовых индексов.

1.7.3. Архитектура сети AlexNet

Сеть AlexNet была разработана специально для конкурса ImageNet Challenge (ImageNet LargeScale Visual Recognition Object Challenge – ILSVRC, 2012), который состоялся в 2012 г. Разработчики AlexNet (Крижевский и др., 2012) сделали ставку на доработку довольно старой схемы, основанной на CNN. Чтобы достичь успеха, им пришлось радикально улучшить архитектуру CNN, и это неизбежно привело к созданию очень большого программного движка; затем им пришлось значительно ускорить его с помощью графических процессоров – задача не из легких, поскольку это означало повторную оптимизацию программного обеспечения в соответствии с оборудованием; наконец, им нужно было придумать, как снабдить модель очень большим обучающим набором – опять же непростая задача, поскольку нужно было тщательно обучить беспрецедентно большое количество параметров, и для этого потребовалось несколько нововведений.

В архитектуре CNN было 10 скрытых уровней (считая уровни C, F и S) – всего на 4 больше, чем у LeNet. Однако эти числа вводят в заблуждение, так как общая глубина различных слоев в AlexNet составляет 11 176 по сравнению с 258 у LeNet. Точно так же AlexNet содержит 650 000 нейронов по сравнению с 6508 у LeNet, а количество обучаемых параметров составляет около 60 млн по сравнению с 60 000 у LeNet. И когда мы смотрим на размер входного изображения, мы обнаруживаем, что AlexNet получает цветное изображение размером 224×224 , тогда как LeNet может обрабатывать только двухуровневое входное изображение 32×32 . Таким образом, в целом AlexNet больше, чем LeNet, в 100–1000 раз, в зависимости от того, какие параметры считать наиболее значимыми. Однако ключевым изменением, внесенным AlexNet, стала возможность работать с огромным количеством слоев и, несмотря на это, решать проблему назначения коэффициентов, при этом все еще исполь-

зую алгоритм обратного распространения ошибки для обучения. В то время это было беспрецедентным достижением, но отчасти это стало возможным благодаря уменьшенному количеству параметров, необходимых для CNN, потому что все нейроны в любом заданном слое нейронов имеют идентичные параметры; это также стало возможным благодаря использованию исключительно больших обучающих наборов. Однако уникальной особенностью архитектуры AlexNet является горизонтальное разделение всей сети на две части, причем для реализации верхней и нижней частей используются разные графические процессоры. В принципе, это должно было чрезмерно усложнить работу с архитектурой, но на практике оказалось выполнимым. Но из-за этой сложности нам будет гораздо проще сосредоточиться на архитектуре ZFNet (Zeiler, Fergus, 2014), так как это, по сути, слегка доработанная и улучшенная версия AlexNet, реализованная на одном графическом процессоре (рис. 1.32): в частности, у нее было восемь, а не семь скрытых слоев (здесь S-слои считаются частью соответствующих C-слоев). Также стоит отметить, что ZFNet выполняет более плавное начальное сужение размеров слоя ($n \times n$), чем AlexNet (уровень C1 имеет размер 110×110 , а не 55×55). С другой стороны, обе архитектуры использовали «перекрывающийся пулинг» – в данном случае комбинацию пулинга 3×3 и страйда 2×2 . Обратите внимание, что размеры ($n \times n$) слоев начинаются с 224×224 и постепенно уменьшаются до 1×1 . Интересно, что почти все обучаемые параметры находятся в слоях F7 и F8 (F6 и F7 для AlexNet), а для окончательного классификатора softmax (не нейронного) остается всего 1000 связей.

На рис. 1.32 слои S1, S2 и S3 показаны синим цветом справа от C1, C2 и C5 соответственно; $n \times n$ означает размеры в случае двумерного формата изображения; $r \times r$ – размер поля ввода двумерного нейрона (одиночное число указывает на абстрактные одномерные данные); $s \times s$ – двумерный страйд. N – это глубина в пределах отдельного слоя: его приблизительный размер указан на рисунке в вертикальном масштабе.

Незадолго до завершения AlexNet, Хинтон с коллегами представили новую методику, называемую «отсев» (dropout). (Hinton et al., 2012; см. также Hinton, 2002). Цель этого подхода заключалась в том, чтобы сократить случаи переобучения. Искомый результат был достигнут путем случайной установки доли (обычно до 50 %) весов на ноль для каждого шаблона обучения; эта довольно неожиданная техника, судя по всему, работает весьма неплохо: она предотвращает чрезмерную зависимость скрытых слоев от конкретных данных, поступающих к ним. Крижевский и его коллеги (Krizhevsky et al., 2012) включили эту функцию в AlexNet. Выход каждого нейрона случайным образом устанавливается равным нулю с вероятностью 0,5. Это делается перед прямой передачей входных данных, и затронутые нейроны не участвуют в последующем обратном распространении. На следующем прямом проходе другой набор выходов нейронов обнуляется с вероятностью 0,5, и снова затронутые нейроны не участвуют в обратном распространении; аналогичное действие выполняется для всех последующих проходов. Во время тестирования происходит альтернативная процедура, когда все выходы нейронов умножаются на 0,5. Фактически умножение всех выходных сигналов нейронов на 0,5 является приближением к получению среднего геометрического

всех распределений вероятностей выходных сигналов локальных нейронов и основано на том факте, что среднее геометрическое не слишком далеко от среднего арифметического. Отсев был включен в первые два слоя AlexNet и значительно уменьшил переобучение из-за слишком малого количества обучающих данных.

Сеть AlexNet была обучена с использованием 1,2 млн изображений, доступных в рамках задачи ImageNet ILSVRC, и это число является подмножеством полных 15 млн в базе данных ImageNet. Фактически набор ILSVRC-2010 был единственным подмножеством, для которого были доступны тестовые метки; в каждой из 1000 категорий было около 1000 изображений. Однако выяснилось, что этих изображений слишком мало, чтобы обучить CNN той сложности, которая требуется для выполнения точной классификации этой огромной задачи. Поэтому нужно было как-то расширить набор для надлежащего обучения AlexNet и достижения уровня ошибок классификации в интервале от 10 % до 20 %.

При обучении модели были предложены и реализованы два основных способа расширения набора данных. Один заключался в том, чтобы применить к изображениям реалистичные переносы и отражения, дабы создать больше изображений того же типа. Преобразования расширились даже до извлечения пяти участков 224×224 и их горизонтальных отражений из исходных изображений ImageNet 256×256 , что дало в общей сложности по десять участков на изображение. Другой метод заключался в изменении интенсивности и цвета входных изображений. Чтобы сделать это упражнение более строгим, оно было выполнено с использованием анализа основных компонентов (PCA) для определения основных компонентов цвета для набора данных ImageNet, а затем для генерации случайных величин, на которые умножаются собственные значения, тем самым создавая реалистичные вариации исходного изображения. Вместе эти два подхода смогли достоверно обобщить и увеличить размер исходного набора данных в 2000 раз – принцип расширения набора заключался в том, чтобы генерировать реалистичные изменения положения, интенсивности и цвета.

На данном этапе следует подчеркнуть, что цель конкурса заключалась в том, чтобы найти наилучшую модель компьютерного зрения (с наименьшим количеством ошибок классификации), которая способна распознавать образцы блохи, собаки, автомобиля или другого типичного объекта в любой локации на изображении и в любой разумной позе. Более того, машина должна расставлять приоритеты в своих классификациях, чтобы давать по крайней мере пять наиболее вероятных интерпретаций с указанием ожидаемой степени достоверности. Затем каждую машину можно оценить не только по точности ее наилучшей классификации, но и по тому, входит ли классифицируемый объект в топ-5 классификаций машины. Модель AlexNet смогла достичь показателя ошибок 15,3 %, по сравнению с 26,2 % у модели, занявшей второе место. Еще одним значимым событием стало резкое падение уровня ошибок нейросетевых моделей до уровня ниже 20 % для такого упражнения, что означало новый этап в жизни нейронных сетей и привлекло к ним всеобщее внимание.

Стоит отметить, что выдающийся результат был достигнут не только за счет разработки выигрышной архитектуры и создания правильного набора данных для адекватного обучения нейросети, но также за счет сокращения времени обучения до приемлемого уровня. В этом отношении решающую роль сыграло использование GPU. Даже с парой графических процессоров непрерывное круглосуточное обучение заняло примерно неделю. Без графических процессоров на это ушло бы примерно в 50 раз больше времени – скорее всего, около года, – поэтому модель просто опоздала бы на конкурс! (Оценка альтернативной продолжительности основана на том, что GPU работает приблизительно в 50 раз быстрее по сравнению с обычным средним процессором.)

Наконец, следует отметить, что графические процессоры обеспечивают очень хорошую реализацию нейросетевых вычислений из-за их внутреннего параллелизма и, следовательно, их способности обрабатывать большие наборы данных за меньшее количество циклов. К счастью, каждый слой CNN полностью однороден и поэтому идеально подходит для параллельной обработки. Также немаловажен тот факт, что графические процессоры хорошо приспособлены к параллельной работе, поскольку они могут напрямую считывать и записывать данные в память друг друга напрямую, избегая необходимости перемещать данные через память центрального процессора.

1.7.4. Архитектура сети VGGNet Симоняна и Зиссермана

В условиях остающейся нехватки знаний о форме идеальной архитектуры Симонян и Зиссерман (Simonyan, Zisserman, 2015) решили определить эффект дальнейшего увеличения глубины. Для этого они значительно сократили количество параметров в базовой сети, ограничив максимальное поле ввода нейрона до 3×3 . Фактически они ограничили поле ввода свертки и страйд до 3×3 и 1×1 соответственно и установили для поля ввода и страйда каждого слоя подвыборки значение 2×2 . Кроме того, они организовали систематическое и быстрое схождение последовательных слоев от 224×224 вниз до 7×7 в 5 этапов с последующим переходом к 1×1 за один полносвязный этап; затем последовали еще два полносвязных слоя и последний выходной слой softmax (рис. 1.33). Все скрытые слои включали этап нелинейности ReLU (на рисунке не показан). Помимо N «каналов», пять сверточных слоев C1–C5 содержали соответственно 2, 2, 3, 3, 3 идентичных подслоя (на рис. 1.33 не отмечены). Наконец, следует отметить, что в целях эксперимента Симонян и Зиссерман разработали 6 вариантов архитектуры VGGNet с 11–19 взвешенными скрытыми слоями: здесь мы рассматриваем только конфигурацию D (с 16 взвешенными скрытыми слоями), для которой количества одинаковых подслоев в слоях C1–C5 указаны выше, а слои F6, F7 и F8 содержат по 1 взвешенному подслою. Очевидно, что количество взвешенных слоев сильно влияет на количество параметров.

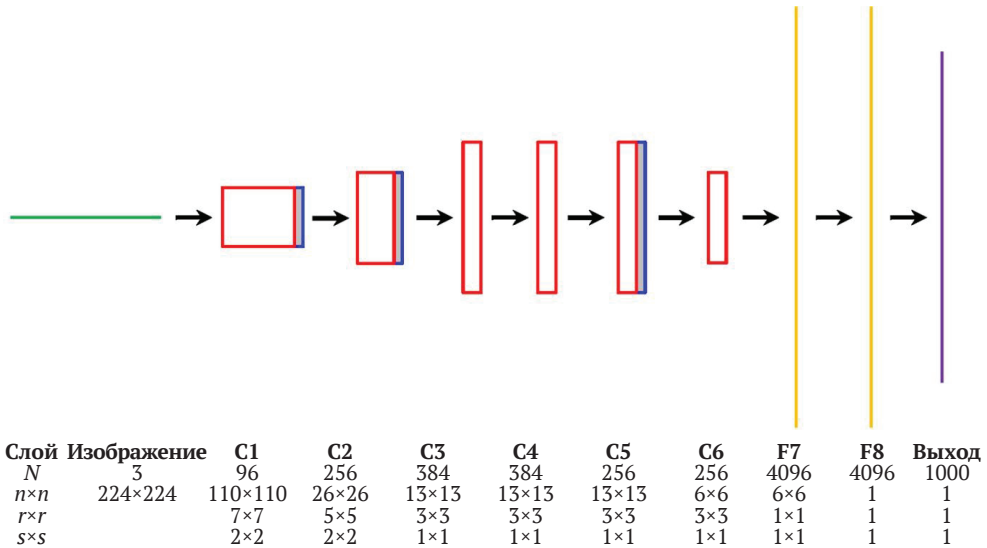


Рис. 1.32 ❖ Схема архитектуры ZFNet. Эта схема очень похожа на схему AlexNet. Обратите внимание, что AlexNet содержит 7 скрытых слоев, тогда как ZFNet содержит 8 скрытых слоев (здесь S-слои считаются частями соответствующих C-слоев). Также отметим, что ZFNet реализована с использованием только одного графического процессора и ее архитектура не разделена

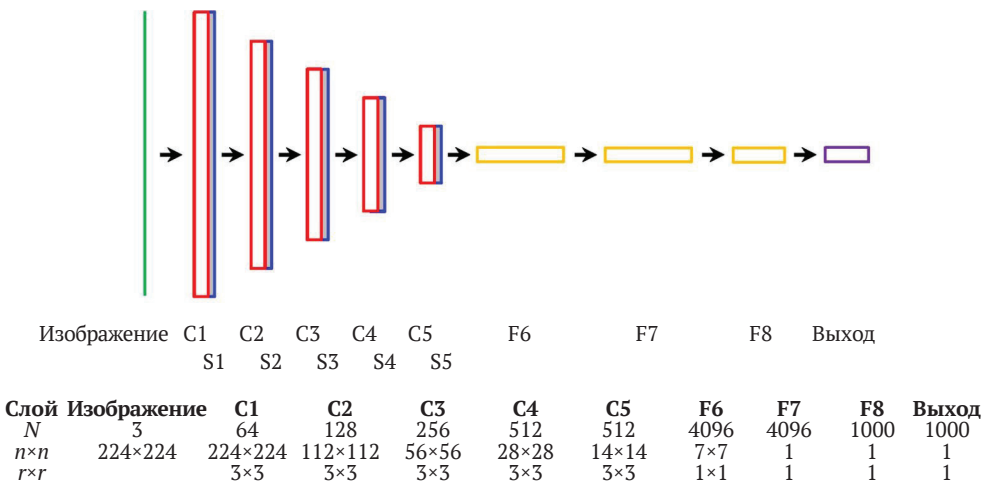


Рис. 1.33 ❖ Архитектура сети VGGNet. Эта архитектура демонстрирует более позднюю оптимизацию стандартной сети CNN. В отличие от схемы на рис. 1.32, здесь показаны относительные размеры слоев свертки, которые варьируются от размера изображения до 1×1 . Обратите внимание, что все слои свертки имеют единичный страйд и что их поля ввода ограничены максимальным размером 3×3 : все слои подвыборки имеют поля ввода 2×2 и страйды 2×2

Как упоминалось выше, Симонян и Зиссерман сократили количество основных параметров, ограничив поле ввода свертки до 3×3 . Это означало, что более крупные свертки должны были быть созданы путем последовательного применения нескольких сверток 3×3 . Ясно, что поле ввода 5×5 потребует применения двух сверток 3×3 , а поле 7×7 потребует трех сверток 3×3 . В последнем случае это уменьшит общее количество параметров с 72×49 до 3 раз по 32×27 . На самом деле этот способ реализации свертки 7×7 не только уменьшил количество параметров, но и потребовал дополнительной регуляризации свертки, поскольку нелинейность ReLU была вставлена между каждой из 3×3 компонентных сверток. Также важно, что и на входе, и на выходе каждого 3-слойного стека свертки 3×3 может быть N каналов, и в этом случае он будет содержать всего $27N^2$ параметров, и именно это число следует сравнивать с $49N^2$ параметрами.

Несмотря на увеличенную глубину, VGGNet содержит только приблизительно в 2,4 раза больше параметров, чем AlexNet; кроме того, она намного проще и не делится на две части с использованием двух GPU. Напротив, она сразу же получает ускорение в 3,75 раза по сравнению с одним графическим процессором при использовании готовой системы с 4 графическими процессорами.

Детали методики обучения аналогичны методике AlexNet, о чем говорится в оригинальной статье Симоняна и Зиссермана (2015). Тем не менее эти авторы вносят одно интересное новшество: это использование «дрожания масштаба» во время обучения, т. е. увеличение обучающей выборки с использованием объектов в широком диапазоне масштабов. На практике было применено случайное масштабирование изображения с коэффициентом, равным 2.

В результате сеть VGGNet достигла показателя 7,0 % ошибок при тестировании с использованием одной сети по сравнению с 7,9 % для GoogLeNet (Szegedy et al., 2014). На самом деле GoogLeNet достигла показателя в 6,7 %, но только за счет использования 7 сетей. Благодаря этому VGGNet заняла второе место в конкурсе ILSVRC-2014. Однако уже после отправки модели на конкурс авторам удалось снизить частоту ошибок до 6,8 %, используя ансамбль из 2 моделей – практически такое же качество классификации, как и у GoogLeNet, но со значительно меньшим количеством сетей. Интересно, что этот результат был получен даже притом, что архитектура VGGNet не отличалась от классической архитектуры LeNet Яна Лекуна (LeCun et al., 1989). Главное улучшение заключается в значительном увеличении глубины сети.

Несмотря на то что сеть VGGNet заняла второе место в конкурсе ILSVRC-2014, она оказалась более универсальной и адаптируемой к различным наборам данных и является предпочтительным выбором в сообществе специалистов по машинному зрению для извлечения признаков из изображений. По-видимому, это связано с тем, что VGGNet на самом деле предоставляет более робастные признаки даже несмотря на то, что качество классификации для определенного набора данных оказалось немного ниже. Как мы увидим в следующем разделе, сеть VGGNet взяли за основу Но и соавторы (Noh et al., 2015) для работы над сетями *деконволюции* (deconvolution, обратная свертка).

1.7.5. Архитектура DeconvNet

Вдохновленные работой Цейлера и Фергюса, Но с коллегами создали «обучаемую сеть деконволюции» (DeconvNet), которая в ходе обучения научилась выполнять деконволюцию наборов коэффициентов свертки в каждом слое CNN. Перед детальным изучением их сети важно понять идею, которой руководствовались авторы. Их целью было создание сети *семантической сегментации*. Суть в том, что обычная сегментация изображения направлена на определение границ между различными объектами, появляющимися на изображении: семантическая же сегментация идет дальше и классифицирует все объекты, тем самым придавая соответствующее значение (семантику) каждой области изображения.

Архитектура DeconvNet показана на рис. 1.34: обратите внимание, что ее начальная секция CNN заимствована из слоев C1–F7 VGGNet, хотя она исключает слой F8 и выходной слой softmax. Будет полезно понять причину этого решения. Во-первых, для распознавания объектов на входном изображении нужна восходящая CNN. Во-вторых, если объекты должны быть расположены в определенных частях изображения, необходима другая CNN, чтобы указать на позиции, и она обязательно должна следовать за процессом распознавания. Выполнение обеих задач в одной огромной неограниченной CNN будет переполнять память и препятствовать обучению, поэтому две сети должны быть тесно связаны друг с другом. Под связью сетей здесь подразумевается

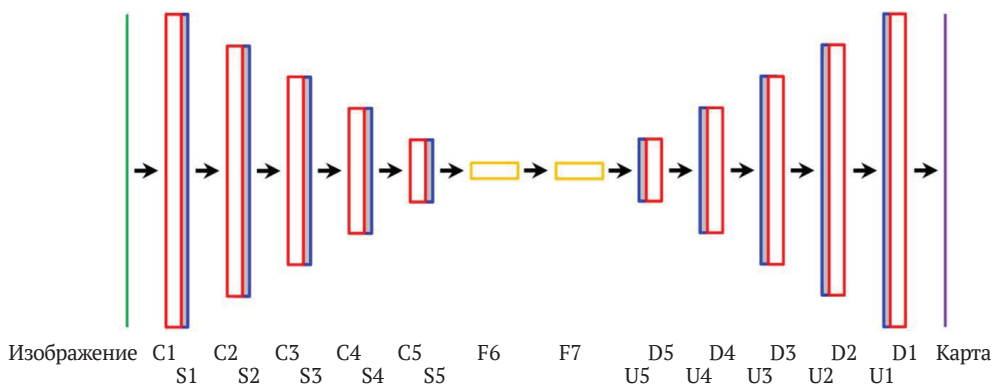


Рис. 1.34 ❖ Схема обучающей сети деконволюции Но. Эта сеть фактически содержит две смыкающиеся сети. Слева – стандартная сеть CNN, а справа – соответствующая сеть «деконволюции» DNN, которая, очевидно, работает в обратном порядке. Сеть CNN (слева) не имеет выходного (например, softmax) классификатора, поскольку конечной целью является не классификация объектов, а представление попиксельной карты их расположения по всей области изображения. Слои деконволюции от D5 до D1 предназначены для постепенного «разделения» слоев C5–C1. Аналогичным образом разделение слоев с U5 по U1 предназначено для постепенного разделения объединенных слоев S5–S1. Для этого параметры положения из слоев max-пулинга должны передаваться в соответствующие местоположения в нужных слоях разделения (т. е. местоположения из S_j должны передаваться в U_i)

обеспечение путей прямой связи от блоков пулинга к последующим блокам разделения. Таким образом, за средствами, с помощью которых выходные данные CNN были обобщены для устранения влияния вариаций выборки, следует вторая CNN, которая дополняет систему для получения необходимых карт местоположения. Важно отметить, что общий восходящий путь данных делает очевидным, почему все блоки ReLU теперь должны указывать в одном направлении. (Теперь все они снова направлены вперед.) Также ясно, что с такой огромной сетью обучение должно проводиться аккуратно, и кажется очевидным, что восходящая часть, отвечающая за обнаружение объектов, должна изначально обучаться самостоятельно.

В целом система работает, зеркально отражая входную CNN путем включения после нее сети деконволюции (DNN). Операция может быть резюмирована следующим образом: слой разделения U_i является нелинейным и перенаправляет (разделяет) максимальные сигналы C_i ; затем слой деконволюции D_i работает с данными линейно и, следовательно, должен суммировать перекрывающиеся входные данные, взвешенные по мере необходимости. Однако вместо того, чтобы создавать подходящие комбинированные правила для определения того, что происходит с перекрывающимися окнами вывода каждого слоя D_i , – и делать это каким-то очень приблизительным образом (например, брать «транспонированные» версии фильтров свертки), – слой деконволюции обучается как обычные части общей сети. Поскольку это строгий подход, он существенно увеличивает нагрузку, связанную с обучением сети.

Читателям будет полезно построить мысленную модель всего процесса, происходящего в DNN. Во-первых, каждый слой разделения восстанавливает информацию из соответствующего слоя пулинга и восстанавливает размеры пространства данных, которые были до объединения. Однако он заполняет пространство только *разреженно*, с локальными максимальными значениями в соответствующих позициях. Целью следующего слоя деконволюции является реконструкция плотной карты в ее пространстве данных. Таким образом, в то время как CNN уменьшает размер активаций, следующая DNN увеличивает активации и снова делает их плотными. Тем не менее полного обращения не происходит, так как повторно вставляются только максимальные значения. Как авторы архитектуры говорят в своей статье (2015): «Разделение захватывает *характерные для образца* (example-specific) структуры, прослеживая исходные местоположения с сильными активациями до пространства изображения», тогда как «обученные фильтры в слоях деконволюции склонны захватывать очертания, *характерные для класса* (class-specific)». Это означает, что слои деконволюции перестраивают очертания образцов, чтобы они более точно соответствовали тому, что можно было бы ожидать для объектов определенных классов.

Несмотря на эту уверенность, сеть должна быть обучена соответствующим образом. Однако сделанное выше предположение о двухэтапном обучении было уточнено авторами работы следующим образом: чтобы решить проблему чрезвычайно большого пространства семантической сегментации, сеть сначала обучается на простых примерах, а затем обучается на более сложных примерах: это равносильно методу начальной загрузки. Точнее, начальный

процесс обучения включает в себя ограничение изменений размера и местоположения объектов путем их центрирования и обрезки в их ограничивающих прямоугольниках; второй этап включает в себя обеспечение того, чтобы более сложные объекты адекватно перекрывались с достоверной сегментацией: для этого используется широко используемая мера *пересечения над объединением* (intersection over union, IoU), которая считается приемлемой, только если она составляет не менее 0,5. На самом деле на первом этапе используется «узкая» ограничивающая рамка, которая увеличивается в 1,2 раза и расширяется до квадрата, чтобы включить достаточный локальный контекст вокруг каждого объекта. На этом первом этапе поле оценивается в соответствии с объектом, расположенным в его центре, а остальные пиксели помечаются как фон. Однако на втором этапе это упрощение не применяется и для аннотации используются все соответствующие метки классов.

Далее мы рассмотрим другой близкий метод, предложенный Бадринарайаной и соавторами (Badrinarayanan et al., 2015), который использует гораздо меньше памяти и имеет ряд других преимуществ.

1.7.6. Архитектура SegNet

Архитектура SegNet сильно напоминает DeconvNet (рис. 1.34) и также нацелена на семантическую сегментацию. Однако ее авторы продемонстрировали необходимость возврата к значительно более простой архитектуре, чтобы сделать ее более легко обучаемой (Badrinarayanan et al., 2015). В основном она была идентична DeconvNet (рис. 1.34), но за исключением слоев F6 и F7. Кроме того, авторам было ясно, что использование max-пулинга и подвыборки снижает разрешение карты признаков и тем самым снижает точность определения местоположения на окончательных сегментированных изображениях. Тем не менее они начинают с устранения полносвязных слоев VGGNet, сохраняя структуру кодирования-декодирования (CNN-DNN) DeconvNet, а также сохраняя max-пулинг и разделение. Фактически именно отказ от использования полносвязных слоев больше всего помогает SegNet, поскольку это резко сокращает количество параметров, которые необходимо изучить, и тем самым так же резко снижает требования к процессу обучения. Соответственно, всю сеть можно рассматривать как единую, а не двойную сеть, и эффективно обучать ее «от начала до конца». Кроме того, авторы определили гораздо более эффективный способ хранения информации о местоположении объекта: они делают это, сохраняя *только* индексы max-пулинга, а именно расположение максимальных значений признаков в каждом окне пула на каждой карте признаков кодировщика. В результате для каждого окна пулинга 2×2 требуется только 2 бита информации (рис. 1.33). Это означает, что даже для начальных слоев CNN (кодировщика) нет необходимости хранить сами карты признаков: необходимо хранить лишь информацию о местоположении объекта. Это означает, что требования к памяти кодировщика снижаются со 134 МБ (соответствует уровням C1–F7 VGGNet) до 14,7 МБ. Общий объем памяти для SegNet будет в два раза больше, поскольку такой же объем информации должен храниться в слоях декодера.

Однако то же самое можно сказать и о других сетях деконволюции, поэтому во всех случаях общий объем данных должен быть удвоен по отношению к содержимому исходного кодировщика CNN.

Меньший размер SegNet делает возможным сквозное обучение и, следовательно, гораздо больше подходит для приложений реального времени. Авторы признают, что более крупные сети могут работать лучше, хотя и за счет гораздо более сложных процедур обучения, увеличения памяти и значительного увеличения времени вывода. Кроме того, трудно оценить их истинную точность. По сути, декодеры должны обучаться с помощью очень больших и громоздких кодировщиков, а последние являются универсальными, а не ориентированными на конкретные приложения. В большинстве случаев такие сети были основаны на внешнем интерфейсе VGGNet, обычно содержащем все 13 подуровней C1–C5 вместе с переменным (очень небольшим) количеством полностью связанных уровней.

На этом фоне неудивительно, что Бадринараяна с соавторами успешно применили SegNet к набору данных CamVid (Brostow et al., 2009 г.), обучив его от начала до конца для оптимальной адаптации. Они обнаружили, что SegNet превзошла семь традиционных (не нейронных) методов, включая дескрипторы локальных меток и суперанализ (Yang et al., 2012; Tighe and Lazebnik, 2013), получив средний результат 80,1 % по сравнению с 51,2 % и 62,0 % соответственно; были распознаны 11 категорий: здание, дерево, небо, транспортное средство, знак, дорога, пешеход, забор, столб, тротуар и велосипедист, – и точность, достигнутая для этих объектов, варьировалась от 52,9 % (велосипедист) до 94,7 % (тротуар). Об их успехе в решении этой задачи можно судить по результатам их онлайн-демонстрации (<http://mi.eng.cam.ac.uk/projects/segnet/>), которая использовалась для создания изображений на рис. 1.35. Фактически в этой демонстрации пиксели размещены в двенадцати категориях, включая дорожную разметку в дополнение к одиннадцати, перечисленным выше.

Они также провели тщательное сравнение SegNet с другими сетями семантической сегментации, включая FCN (fully convolutional networks – полностью сверточные сети) и DeconvNet. Сети FCN и DeconvNet имеют одинаковый размер кодировщика (134M); отметим, что FCN уменьшает размер декодера до 0,5M, хотя DeconvNet продолжает использовать декодер размером 134M. Средние результаты по классам для трех методов составляют 59,1 %, 62,2 % и 69,6 %. Несмотря на то что SegNet численно является худшей из трех, ее точность по-прежнему конкурентоспособна, и у нее есть явное преимущество в том, что она лучше адаптируется благодаря сквозному обучению. Фактически она также является самой быстрой – в 2,2 раза быстрее, чем FCN, и в 3,3 раза быстрее, чем DeconvNet, хотя и на изображениях разного размера.

В целом авторы заявляют, что архитектуры, «которые хранят сетевые карты признаков кодировщика в полном объеме, работают лучше, но потребляют больше памяти во время вывода», что также означает, что они работают медленнее. С другой стороны, SegNet более эффективна, поскольку хранит только индексы max-пула; кроме того, она обладает конкурентоспособной точностью, а ее способность к сквозному обучению на актуальных данных делает ее значительно более адаптируемой.



Рис. 1.35 ❖ Две дорожные сцены, снятые с переднего пассажирского сиденья. В каждом случае изображение слева является исходным, а изображение справа является сегментацией, созданной SegNet. Ключ указывает 12 возможных значений, присвоенных SegNet. Хотя точность определения местоположения не идеальна, присваиваемые значения, как правило, разумны, учитывая ограниченное количество допустимых интерпретаций и разнообразие объектов в поле зрения. Эти изображения были обработаны с использованием онлайн-демонстрации по адресу <http://mi.eng.cam.ac.uk/projects/segnet/> (Badrinarayan et al., 2015)

1.7.7. Применение глубокого обучения для отслеживания объектов

Теперь вернемся к теме слежения за движущимися объектами. Это область, в которой методы глубокого обучения позволили достичь больших успехов. Фактически глубокое обучение привело к радикальным улучшениям по сравнению с традиционными подходами, обсуждавшимися ранее. В качестве интересного примера начнем со статьи Хелда (Held et al., 2016).

Хелд и соавторы поставили перед собой задачу создать трекер одного объекта, который действовал бы в режиме реального времени, и разработали

нейронный метод, работающий на скорости до 100 кадров в секунду. Эта скорость была достигнута только для тестовой версии сети и опиралась на очень значительный объем автономного обучения. Высокая скорость тестирования является результатом использования относительно простой архитектуры, в которой пары кадров подаются на обученную нейронную сеть, которая немедленно (т. е. за один проход) выдает выходное изображение, в котором промаркирована рамка целевого объекта.

При тестировании трекер инициализируется начальной ограничительной рамкой, содержащей целевой объект, причем ограничительная рамка обновляется после анализа каждой последующей пары кадров. Однако перед детальным изучением каждый кадр обрезается до размера и положения, достаточных для захвата целевого объекта при любом движении, которое он может совершить в разумных пределах (обычно это означает кадрирование до размера, в два раза превышающего ограничивающую рамку): эта процедура также позволяет использовать полезную контекстную информацию, которая заслуживает внимания. Затем обученная нейронная сеть ищет два кадрированных изображения ($t - 1$ и t), чтобы найти наилучшее соответствие для положения движущегося объекта. Таким образом, повторение процесса позволяет отслеживать целевой объект на протяжении всей видеопоследовательности.

Для успешного отслеживания обученная сеть должна содержать огромное количество информации о различных возможных смещениях каждой пары изображений. Это вполне возможно при использовании пары сетей, каждая из которых содержит N слоев свертки (часто называемой «сиамской ConvNet»), которые подаются на набор из M полносвязных слоев (Хелд и соавторы использовали $N = 5$ и $M = 3$); окончательный вывод содержал необходимую информацию о выходной ограничивающей рамке. Для получения всей необходимой информации нейросеть обучалась на всех допустимых сдвигах и обрезках поступающих кадров. На самом деле сеть обучалась не только на видео, но и на парах изображений, чтобы научить ее отслеживать более разнообразный набор объектов, тем самым помогая предотвратить переобучение объектам в видеоданных.

В результате тщательного обучения трекер оказался инвариантным к фоновому движению, вращению вне плоскости, деформациям, изменениям освещения и незначительным окклюзиям. Кроме того, дополнительное обучение на размеченных изображениях помогло установить общую взаимосвязь между внешним видом и движением объекта, что позволило отслеживать объекты, не появляющиеся в обучающем наборе, а также возможность выполнять эту задачу с беспрецедентной скоростью 100 кадров в секунду.

Однако за это пришлось расплачиваться большой продолжительностью обучения из-за чрезмерного увеличения объема данных. Это была одна из проблем, побудивших Бертинетто и др. (Bertinetto et al., 2016) к разработке архитектуры отслеживания с использованием *полностью сверточной* (fully convolutional, FC) *сиамской сети*, которую они называли SiamFC. Они начали с системы, имеющей два параллельных входа, один из которых предоставлял эталонный образец изображения, а другой – входное изображение. Идея состоит в том, чтобы искать во входном изображении совпадения с эталонным

изображением, используя подходящий оператор подобия. Фактически для этой цели они использовали корреляцию, а корреляционный слой применялся для создания карты оценок, которая была представлена на выходе сети. Разработчики решили сделать так, чтобы карта выходных оценок имела уменьшенную размерность, из которой попадания можно было бы соотнести обратно с входным изображением путем умножения на сетевой страйд. Последний был принят равным 8 – значение, которое возникает, когда есть три последовательных применения шага 2×2 в слоях C1, S1 и S2 (рис. 1.36). Чтобы полностью понять общий эффект, обратите внимание, что каждый страйд 2×2 делит собственный размер изображения на 2×2 .

Сиамская сеть состоит из двух параллельных потоков, каждый из которых имеет одинаковую внутреннюю структуру. Это естественно для описанной выше архитектуры, поскольку два потока несут данные одного и того же типа, хотя размеры изображений, очевидно, будут разными. Для каждого из потоков используется полностью сверточная сеть, так как цель состоит в достижении трансляционной инвариантности, чтобы сигнал от любого объекта не менялся в зависимости от его положения на изображении. «Полностью сверточная» означает, что никакие изменения параметров свертки не допускаются во всем пространстве изображения; иными словами, это означает, что паддинг, т. е. «заполнение» внешних участков сети (обычно нулями), не допускается, и на практике это означает, что – в отличие от AlexNet – выходные пространства последовательных слоев будут сокращены везде, где входные поля нейрона превышают 1×1 (рис. 1.36). Следует добавить, что смысл включения двух полностью сверточных сетей состоит в том, чтобы получить достаточную информацию о наблюдаемом объекте и объекте-образце, игнорируя при этом шум, изменения формы и несущественные артефакты, такие как детали фона.

Корреляция выполняется с использованием стандартного вычисления скользящего окна. Общая система показана на рис. 1.36; две полностью сверточные потоковые сети адаптированы из архитектуры AlexNet Крижевского и др. (2012). Заметим, что полностью связанные слои были отброшены, поскольку они не являются полностью сверточными и поэтому не допускают позиционной инвариантности. (Цель AlexNet заключалась в том, чтобы классифицировать любое входное изображение в соответствии с классом целевого объекта, появляющегося в нем, при этом информация о локализации объекта полностью отбрасывалась.)

В работе Бертинетто и др. эталонный образец принимается за начальный вид целевого объекта и не обновляется; также не сохраняется память о прошлых явлениях и не вычисляются прогнозы положения ограничивающих рамок объекта. Главная цель состояла в том, чтобы создать простой и надежный трекер для отдельных целевых объектов. Однако упомянутые исследователи обнаружили, что повышение дискретизации карты оценок в 16 раз с использованием бикубической интерполяции (т. е. с 17×17 до 272×272) дало более точную локализацию. Кроме того, они искали объект в пяти масштабах, от 0,95 до 1,05, чтобы справиться с вариациями масштаба.

Далее мы рассматриваем размер выходного пространства изображения. Слой корреляции представляет собой свертку между изображениями разме-

ром 22×22 и 6×6 , что приводит от потенциально максимального выходного размера изображения 27×27 к приемлемому размеру изображения 17×17 (эти числа возникают из-за того, что $22 + 6 - 1 = 27$ и $22 - 6 + 1 = 17$). Обратите внимание, что максимальный размер выходного изображения (27×27) не будет захватывать объекты, которые находятся сразу за пределами области поиска, тогда как минимальный (17×17) будет полностью учитывать объекты, сплошь находящиеся в пределах области поиска; между этими пределами существует та или иная вероятность обнаружения частично видимых объектов. Тем не менее SiamFC был нацелен на максимально успешное обнаружение полностью видимых объектов.

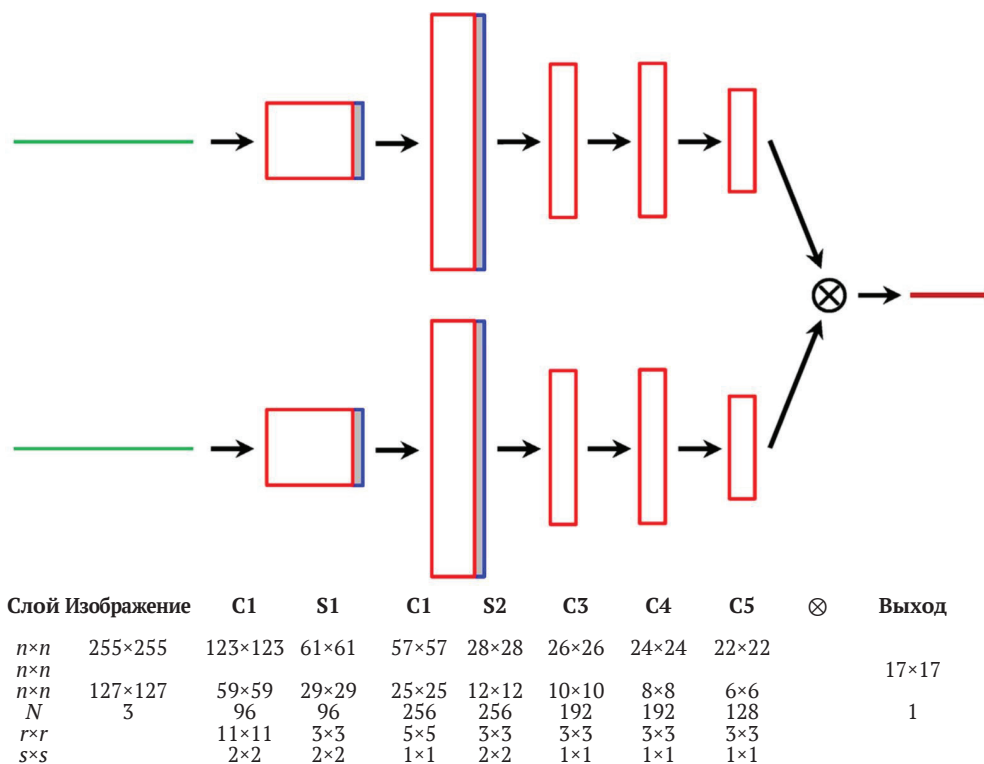


Рис. 1.36 ❖ Полностью сверточный сиамский трекер. Здесь показана архитектура сиамского трекера FC, разработанного Бертинетто и др. (Bertinetto et al., 2016). Вверху: ветвь FC, содержащая поток *поиска*; середина: ветвь, содержащая поток *образцов категории*; внизу: детали двух ветвей, в т. ч. каналов N , размер изображения $n \times n$, входное поле нейрона $r \times r$ и страйд $s \times s$ (N , r и s одинаковы для двух потоков). Обратите внимание, что, как и в случае с AlexNet, в этой архитектуре используется перекрывающийся пул в слоях подвыборки S1 и S2, для обоих из которых $r \times r = 3 \times 3$ и $s \times s = 2 \times 2$

Наконец, интересный момент касается расчета эффектов страйда 2×2 , когда предыдущее изображение имеет нечетный размер (как это происходит во всех трех случаях в SiamFC). В этом случае страйд учитывает первый и по-

следний пиксели в каждой строке и столбце. Принимая это во внимание, последовательные размеры изображений, приведенные на рис. 1.36, точно и логически соответствуют размерам, данным разработчиками (Bertinetto et al., 2016).

Файхтенхофер и др. (Feichtenhofer et al., 2017) задались целью разработать архитектуру глубокого обучения, которая одновременно изучает обнаружение и отслеживание. В их архитектуре используются детектор объектов и трекер: по сути, идея состоит в том, чтобы одновременно выполнять обнаружение и отслеживание с использованием сети ConvNet, оптимизированной за счет комбинированного обнаружения и отслеживания на основе потерь. Для оценки локального смещения в разных масштабах признаков выполняется сверточная взаимная корреляция между откликами признаков соседних кадров. Здесь корреляция ограничена небольшой окрестностью с максимальным смещением $d = 8$, чтобы избежать большой выходной размерности: это отражает ограничение, уже отмеченное для средства отслеживания целей (Held et al., 2016). Карты корреляции вычисляются для всех позиций на карте признаков, и для отслеживания регрессии к этим картам признаков применяется пулинг видимых областей (region of interest, RoI) (Dai et al., 2016). Вышеупомянутый подход позволяет одновременно отслеживать несколько объектов, а архитектуру можно обучать от начала до конца, используя входные кадры из необработанных видео. В целом этот подход привел к значительному повышению качества на уровне 80 % по сравнению с предыдущими современными методами при выполнении на наборе проверки VID ImageNet (2015).

Обратите внимание, что подход пулинга RoI использует полностью сверточную схему, насколько это возможно, чтобы поддерживать сдвиговую инвариантность, и только последний сверточный слой модифицируется, чтобы отклониться от этого: данная стратегия позволяет вычислять карты оценок, чувствительные к положению, из которых могут быть извлечены так называемые *треклеты* (tracklet) нескольких объектов. Однако, поскольку детали общей архитектуры тесно связаны с R-FCN (Dai et al., 2016), ResNet-101 (He et al., 2016), Fast R-CNN (Girshick, 2015) и Faster R-CNN (Ren et al., 2015), доступное место не позволяет привести здесь полное описание.

1.7.8. Применение глубокого обучения в классификации текстур

Как мы видели в предыдущих разделах, глубокое обучение стало основной движущей силой разработки алгоритмов зрения. В равной степени это относится и к анализу текстур, который был представлен в разделах 1.6.1–1.6.6. На рис. 1.31 показана архитектура классификатора текстур Лоуза, рассмотренного в разделе 1.6.3. Классификатор такого типа часто описывается как использующий набор «банков фильтров» для извлечения входной информации. Однако в предыдущем разделе показано, что аналогичным образом могут быть описаны методы CNN, поэтому неудивительно, что CNN (и ANN)

также применялись для анализа текстуры. Раньше такие сети обычно обучали, используя наборы входных изображений, каждое из которых состояло из одной текстуры, обычно из базы данных текстур Бродача (Brodatz textures database). Однако этот подход ограничен обработкой всего входного изображения как одной области и соответствующей его классификацией: сегментация изображения на области с разными текстурами выходит за рамки возможностей простой архитектуры, обученной таким образом.

Андреарчик и Уилан (Andrearczyk, Whelan, 2016) описали базовую архитектуру CNN (T-CNN-3) для классификации текстур, которая показана на рис. 1.37. Она имеет некоторое сходство с ZFNet, хотя содержит меньше слоев свертки; впрочем, это не мешает ей справляться с текстурами, потому что признаки текстуры в большинстве случаев можно описать с помощью довольно маленьких окон. Также обратите внимание, что за последним слоем свертки следует слой объединения, который усредняет энергию текстуры по всей карте объектов. Тем не менее, поскольку значительное количество N карт признаков вычисляется различными параллельными слоями конечного слоя свертки, результатом является вектор из N значений энергии. Они передаются в окончательный классификатор текстур, который дает один класс для любого одного входного изображения текстуры.

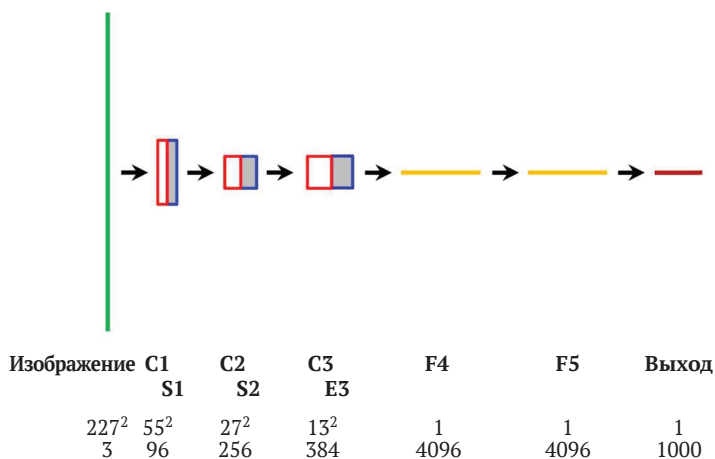


Рис. 1.37 ❖ Архитектура T-CNN Андреарчика и Уилана. Она предназначена для захвата текстуры всего входного изображения и вывода вектора, указывающего на наиболее вероятную текстуру. Цифры под метками указывают размеры пространств признаков, обрабатываемых слоями свертки C1–C3, и количество карт признаков в соответствующих слоях. Оба слоя объединения (S1, S2) уменьшают размер изображения в 2×2 раза. Энергетический слой E3 – это особый слой пулинга, который усредняет энергии полностью по каждой карте признаков, создавая 384 вывода, по одному для каждой карты признаков C3

Фактически Андреарчик и Уилан (Andrearczyk, Whelan, 2016) пошли еще дальше и разработали анализатор текстуры гибридного типа (TS-CNN-3), использующий как описанный выше метод, так и позволяющий анализиро-

вать формы объектов, причем последний действует аналогично системе распознавания объектов AlexNet для нетекстурированных изображений. Общая архитектура показана на рис. 1.38. Важным элементом этой архитектуры является слой конкатенации M, объединяющий выходные данные частей системы, обрабатывающих текстуры и формы. Но также важно отметить, что релевантная информация о текстуре в основном получается из относительно небольших признаков низкого уровня, тогда как информация о форме является более глобальным свойством, которое требует ввода признаков более высокого уровня. Вот почему информация о текстуре получается из выходных данных слоя свертки C3, тогда как информация о форме получается из пулинговых выходных данных C5. (Интересно, что, поскольку E3 усредняет энергию текстуры по всему характеристическому слою C3, в текстурном канале после E3 не остается информации о пространстве или форме.) Наконец, выходные слои F6, F7 и O используются для объединения выходных данных формы и текстуры: ясно, что в этих слоях имеется достаточно информации, чтобы связать текстуры с определенными местами на входном изображении и представить выходные данные с точки зрения вероятностей того, что текстуры, используемые при обучении, действительно появляются на входном изображении. Впрочем, чего эта сеть *не* делает, так это создание карты вы-

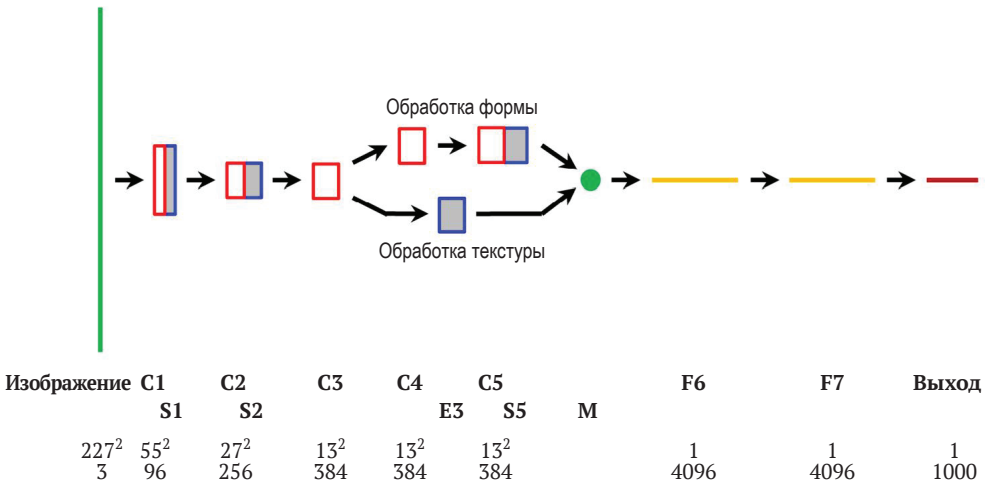


Рис. 1.38 ❖ Архитектура TS-CNN Андреарчика и Уилана. Она предназначена для захвата информации о текстуре и форме для всего входного изображения и для вывода вектора, показывающего наиболее вероятный набор текстур. Первый ряд чисел под метками указывает размеры пространств признаков, обрабатываемых слоями свертки C1–C5; второй ряд чисел указывает номера карт объектов в соответствующих слоях. Все три слоя объединения (S1, S2, S5) уменьшают размер изображения в 2×2 раза, поэтому изображение, подаваемое в точку слияния M, имеет размер 6×6. В M выходные данные текстуры и формы объединяются, в результате чего получается в общей сложности 384×(1 + 6²) выходных данных – все из которых соединены со всеми входами F6, образуя полносвязную сеть. Как и на рис. 1.37, энергетический слой E3 является объединяющим слоем, который усредняет энергию по всей карте признаков, создавая 384 результата, по одному для каждой карты признаков C3

ходного изображения, показывающей наиболее вероятную разбивку изображения на различные текстурные области. Было бы неплохо добиться этого, и в принципе это должно быть возможно за счет включения архитектуры кодер–декодер SegNet (раздел 1.7.6). Основным препятствием здесь является отсутствие достаточно больших наборов текстурных данных (сравнимых с ImageNet): на самом деле основная проблема заключается в том, что такие архитектуры, как SegNet, настолько глубоки, что требуют гораздо большего объема обучения с использованием гораздо большего количества обучающих шаблонов.

Отсутствие большого набора текстурных данных является серьезным фактором, мешающим дальнейшему прогрессу в анализе текстур, какие бы методы обучения и архитектуры ни разрабатывались. И действительно, сегодня уже ясно, что если бы такой набор данных был доступен, AlexNet (и другие сети, такие как VGGNet) можно было бы обучить выполнять анализ текстур по всем изображениям без внесения каких-либо изменений в их архитектуру. Как отмечают исследователи (Liu et al., 2019): «Недавний успех глубокого обучения в классификации изображений и распознавании объектов неотделим от наличия крупномасштабных аннотированных наборов данных изображений, таких как ImageNet. Однако анализ текстур на основе глубокого обучения не поспевает за быстрым прогрессом, наблюдаемым в других областях, отчасти из-за отсутствия крупномасштабной базы данных текстур».

Эти утверждения вполне обоснованы, потому что (а) архитектуры на основе CNN уже являются одними из наиболее широко используемых средств выполнения анализа текстур и вряд ли утратят свои позиции в обозримом будущем; и (б) те же ограничения на обучение (и потребность в больших наборах текстурных данных) справедливы для любых методов, которые будут использоваться для анализа текстурированных изображений. Следует также помнить, что статистический характер текстур подразумевает, что в целом процедуры, основанные на обучении, будут предпочтительнее созданных вручную алгоритмов.

Наконец, сравним производительность двух архитектур Андреарчика и Уилана. При обучении на ImageNet, который представляет собой *объектный* набор данных, они обнаружили, что T-CNN-3 работает хуже, чем AlexNet (51,2 % против 57,1 %), что не удивительно, потому что AlexNet – намного более крупная сеть по сравнению с 23,4 млн T-CNN-3. С другой стороны, при обучении с текстурно-ориентированными наборами данных, такими как ImageNet-T и KTH-TIPS-2b (Russakovsky et al., 2015; Hayman et al., 2004), T-CNN3 показала значительное улучшение по сравнению с AlexNet (соответствующая точность 71,1 % по сравнению с 66,3 % и 48,7 % по сравнению с 47,6 %). Эти результаты соответствуют обучению с нуля на одной базе данных. Однако также можно провести предварительную подготовку в одной базе данных и выполнить точную настройку в другой. Когда это было предпринято (в частности, выполнялось предварительное обучение на ImageNet и точная настройка на KTH-TIPS-2b), сеть T-CNN-3 работала точнее, чем AlexNet (73,2 % по сравнению с 71,5 %). Дальнейшее улучшение было достигнуто за счет использования гибридной архитектуры TS-CNN-3, при этом была достигнута точность 74,0 %. Частично это улучшение явно было

связано с большим количеством обучаемых параметров TS-CNN-3 (62,5 млн). Тем не менее, когда были проведены испытания архитектур с сопоставимым количеством параметров, например комбинация AlexNet и T-CNN-3 и комбинация VGG-M (Chatfield et al., 2014) с FV-CNN (Cimpoi et al., 2015), сеть TS-CNN-3 осталась лучшей. Фактически отличительной чертой новой архитектуры TS-CNN-3 является то, что она очень эффективно использует параметры обучения, а за счет отделения анализа текстуры от анализа формы достигаются более точные результаты при обработке текстурированных изображений.

Мы не напрасно уделили такое внимание работе Андреарчика и Уилана, потому что она раскрывает множество идей о применении глубоких сетей для анализа текстур, включая ряд связанных с ними тонкостей. В то время как их подход нацелен на архитектуру с относительно небольшим количеством обучающих параметров, существует менее ограниченный подход (Cimpoi et al., 2015; 2016), который кажется более мощным: он систематически использует глобальный пулинг CNN до уровня полностью связанных слоев и, таким образом, захватывает ценные наборы дескрипторов текстуры. (Точнее, он выполняет более плотный пулинг локальных признаков, удаляя глобальную пространственную информацию, чтобы извлечь плотные сверточные признаки.) Действительно, в работе (Cimpoi et al., 2016 г.) продемонстрирована разработка словаря из сорока семи текстурных атрибутов, которые описывают широкий выбор шаблонов текстур. Фактически эти работы оказали влияние на почти идеальную точность классификации. Однако Лю и др. (Liu et al., 2019) заявляют, что качество близко к «насыщению» (т. е. стабилизировалось на уровне около 100 %), «поскольку наборы данных недостаточно велики, чтобы можно было выполнить точную настройку для получения улучшенных результатов», что подтверждает ранее сделанные замечания о доступных наборах текстурных данных. Следует отметить, что наилучшие результаты были получены с использованием архитектуры VGG-VD (Very Deep) Симоняна и Зиссермана (Simonyan, Zisserman, 2015), содержащей 19 слоев CNN, что частично объясняет, почему насыщение стало возможным.

Еще один вопрос, поднятый в работе Лю и др. (Liu et al., 2019) в их обзоре развития представлений текстур, заключается в том, что существует растущее противоречие между потребностью сетей в огромных наборах данных изображений и соответствующей человеческой потребностью в компактных, эффективных представлениях. Последняя потребность все чаще проявляется на мобильных и встроенных платформах с ограниченными ресурсами, как ранее отмечали Андреарчик и Уилан (Andrearczyk, Whelan, 2016) и Сегеди и др. (Szegedy et al., 2014).

1.7.9. Анализ текстур в мире глубокого обучения

Часть Е началась с изучения определения текстуры – по сути, с вопроса «Что такое текстура и как она формируется?». Как правило, текстура начинается с участка поверхности, имеющего локальную шероховатость или структуру, который затем многократно проецируется для формирования текстуриро-

ванного изображения. Такое изображение демонстрирует как регулярность, так и случайность, хотя важными параметрами могут быть также направленность и ориентация. Однако наличие важного признака случайности означает, что текстуры должны характеризоваться статистическими методами и распознаваться с использованием процедур статистической классификации. Методы, которые использовались для этой цели, включают автокорреляцию, матрицы совместной встречаемости, меры энергии текстуры, меры на основе фракталов, марковские случайные поля и т. д. Они направлены как на анализ, так и на моделирование текстур. Специалистам в этой области приходилось тратить много времени на создание все более совершенных моделей текстур, с которыми они работают, чтобы лучше описывать, распознавать и сегментировать их. Однако менее чем за десятилетие DNN внезапно обрели самостоятельность, и, как мы убедились, это привело к стремительному развитию нейросетевых методов анализа текстур, что еще больше увело нас от прямых методов 1970-х годов.

1.8. Часть G. ЗАКЛЮЧЕНИЕ

Части А–Е этой главы посвящены трем весьма актуальным темам по обнаружению признаков и объектов как в двухмерных, так и в трехмерных изображениях. В них представлены базовые методы, которые широко применялись в компьютерном зрении до взрыва глубокого обучения в 2012 году. Часть F продолжает рассказ о том, что повлек за собой этот взрыв (в частности, что все признаки и объекты можно выучить на многочисленных обучающих примерах), но также и о том, как с помощью сквозного обучения полных сверточных сетей может быть достигнута семантическая сегментация. Пожалуй, можно считать благотворным тот факт, что такие сети могут изучать все признаки и объекты и, по-видимому, не нуждаются в устаревших методах для их хранения. На самом деле семантическая сегментация статических сцен – не единственный наглядный пример, но, как показано в разделе 1.7.7, похожий процесс с впечатляющим успехом был реализован для отслеживания нескольких движущихся объектов. Действительно, как было указано в одной из статей (Feichtenhofer et al., 2017), многое из того, что раньше достигалось сложными способами путем жесткого применения правил модели, теперь достигается менее трудоемкими и сложными способами с помощью методов глубокого обучения; и они способны поднять качество моделей на еще большие высоты, если приложить дополнительные усилия. Оглядываясь назад, создается впечатление, что исследователи столкнулись с метафорической кирпичной стеной в отношении того, каким образом достаточно точно указать природу «мягких» данных в *реальных* (неидеализированных) изображениях, которые они хотели обработать. Как следствие у них возникло желание хотя бы попробовать альтернативу глубокого обучения.

На этом этапе будет полезно подытожить, что именно способствовало взрыву глубокого обучения. По сути, это было сочетание множества разрозненных факторов. В частности, мы можем указать на: (1) использование CNN

с гораздо более низкой связностью, чем в случае со старыми архитектурами ANN; (2) использование ReLU вместо сигмоидальных выходных функций; (3) применение процедуры «отсева» для ограничения случаев переобучения; (4) использование значительно увеличенных объемов данных изображений для обучения; (5) извлечение еще большего количества фрагментов изображения из изображений для дальнейшего увеличения объема данных, доступных для обучения; (6) применение графических процессоров для выполнения обучения с чрезвычайно высокой скоростью (принято считать, что графический процессор имеет 50-кратное преимущество в скорости по сравнению с типичным хост-процессором). Следует также помнить, что все эти изменения случились более или менее одновременно в 2012 г.

На этом мы закончим наше несколько затянувшееся введение, поскольку в следующих главах приведено много дополнительных сведений о положении дел в довольно сложной и быстро меняющейся отрасли компьютерного зрения.

БЛАГОДАРНОСТИ

Следующий текст и рисунки воспроизведены с разрешения IET: рисунок в тексте и связанный с ним текст в разделе 1.2.7 – из Electronics Letters (Davies, 1999); рис. 1.2 и связанный с ним текст – из Proc. Visual Information Engineering Conf. (Davies, 2005); выдержки из текста – из Proc. Image Processing and its Applications Conf. (Davies, 1997). Рисунок 1.5 и соответствующий текст воспроизведены с разрешения IFS Publications Ltd. (Davies, 1984). Я также хочу отметить, что рис. 1.13 и 1.15 и связанный с ними текст были впервые опубликованы в Proceedings of the 4th Alvey Vision Conference (Davies, 1988b).

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Ade F.*, 1983. Characterization of texture by «eigenfilters». *Signal Processing* 5 (5), 451–457.
- Amit Y., Trouvé A.*, 2007. POP: patchwork of parts models for object recognition. *International Journal of Computer Vision* 75 (2), 267–282.
- Andrearczyk V., Whelan P.*, 2016. Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters* 84, 63–69.
- Badrinarayanan V., Kendall A., Cipolla R.*, 2015. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *arXiv:1511.00561v2 [cs CV]*.
- Bai Y., Ma W., Li Y., Cao L., Guo W., Yang L.*, 2016. Multi-scale fully convolutional network for fast face detection. In: *Proc. British Machine Vision Association Conference*. York, 19–22 September. <http://www.bmva.org/bmvc/2016/papers/paper051/paper051.pdf>.
- Ballard D. H.*, 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* 13, 111–122.

- Beaudet P. R.*, 1978. Rotationally invariant image operators. In: Proc. 4th Int. Conf. on Pattern Recognition. Kyoto, pp. 579–583.
- Bertinetto L., Valmadre J., Henriques J. F., Vedaldi A., Torr P. H. S.*, 2016. Fully-convolutional Siamese networks for object tracking. In: Proc. ECCVWorkshops, pp. 850–865.
- Brostow G., Fauqueur J., Cipolla R.*, 2009. Semantic object classes in video: a high-definition ground truth database. *Pattern Recognition Letters* 30 (2), 88–97.
- Canny J.*, 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 679–698.
- Castañón D. A.*, 1990. Efficient algorithms for finding the k best paths through a trellis. *IEEE Transactions on Aerospace and Electronic Systems* 26 (2), 405–410.
- Chatfield K., Simonyan K., Vedaldi A., Zisserman A.*, 2014. Return of the devil in the details: delving deep into convolutional nets. In: Proc. BMVC, pp. 1–12.
- Cimpoi M., Maji S., Vedaldi A.*, 2015. Deep filter banks for texture recognition and segmentation. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3828–3836.
- Cimpoi M., Maji S., Kokkinos I., Vedaldi A.*, 2016. Deep filter banks for texture recognition, description and segmentation. *International Journal of Computer Vision* 118, 65–94.
- Cootes T. F., Taylor C. J.*, 2001. Statistical models of appearance for medical image analysis and computer vision. In: *Sonka M., Hanson K. M.* (Eds.), Proc. SPIE, Int. Soc. Opt. Eng. USA, vol. 4322, pp. 236–248.
- Cucchiara R., Grana C., Piccardi M., Prati A.*, 2003. Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (10), 1337–1342.
- Dai J., Li Y., He K., Sun J.*, 2016. R-FCN: object detection via region-based fully convolutional networks. In: Proc. NIPS.
- Dalal N., Triggs B.*, 2005. Histograms of oriented gradients for human detection. In: Proc. Conf. on Computer Vision Pattern Recognition. San Diego, California, USA, pp. 886–893.
- Davies E. R.*, 1984. Design of cost-effective systems for the inspection of certain food products during manufacture. In: Pugh, A. (Ed.), Proc. 4th Int. Conf. on Robot Vision and Sensory Controls. London, 9–11 October. IFS (Publications) Ltd, Bedford and North-Holland, Amsterdam, pp. 437–446.
- Davies E. R.*, 1986. Constraints on the design of template masks for edge detection. *Pattern Recognition Letters* 4 (2), 111–120.
- Davies E. R.*, 1988a. A modified Hough scheme for general circle location. *Pattern Recognition Letters* 7 (1), 37–43.
- Davies E. R.*, 1988b. An alternative to graph matching for locating objects from their salient features. In: Proc. 4th Alvey Vision Conf. Manchester, 31 Aug.–2 Sept., pp. 281–286.
- Davies E. R.*, 1988c. Median-based methods of corner detection. In: Kittler, J. (Ed.), Proceedings of the Fourth BPRA International Conference on Pattern Recognition. Cambridge, 28–30 March. In: *Lecture Notes in Computer Science*, vol. 301. Springer-Verlag, Heidelberg, pp. 360–369.

- Davies E. R.*, 1997. Designing efficient line segment detectors with high orientation accuracy. In: Proc. 6th IEE Int. Conf. on Image Processing and Its Applications. Dublin, 14–17 July. In: IEE Conf. Publication, vol. 443, pp. 636–640.
- Davies E. R.*, 1999. Designing optimal image feature detection masks: equal area rule. *Electronics Letters* 35 (6), 463–465.
- Davies E. R.*, 2005. Using an edge-based model of the Plessey operator to determine localisation properties. In: Proc. IET Int. Conf. on Visual Information Engineering. University of Glasgow, Glasgow, 4–6 April, pp. 385–391.
- Davies E. R.*, 2017. *Computer Vision: Principles, Algorithms, Applications, Learning*, 5th edition. Academic Press, Oxford, UK.
- Davies E. R., Bateman M., Mason D. R., Chambers J., Ridgway C.*, 2003. Design of efficient line segment detectors for cereal grain inspection. *Pattern Recognition Letters* 24 (1–3), 421–436.
- Dreschler L., Nagel H.-H.*, 1981. Volumetric model and 3D-trajectory of a moving car derived from monocular TVframe sequences of a street scene. In: Proc. Int. Joint Conf. on Artif. Intell., pp. 692–697.
- Dudani S. A., Luk A. L.*, 1978. Locating straight-line edge segments on outdoor scenes. *Pattern Recognition* 10, 145–157.
- Elgammal A., Harwood D., Davis L.*, 2000. Non-parametric model for background subtraction. In: Proc. European Conf. on Computer Vision. In: LNCS, vol. 1843, pp. 751–767.
- Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A.*, 2007. The Pascal visual object classes challenge 2007. (VOC2007) Results. <http://www.pascalnetwork.org/challenges/VOC/voc2007/>.
- Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A.*, 2008. The Pascal visual object classes challenge 2008. (VOC2008) Results. <http://www.pascalnetwork.org/challenges/VOC/voc2008/>.
- Everingham M., Zisserman A., Williams C. K. I., Van Gool L.*, 2006. The Pascal visual object classes challenge 2006. (VOC2006) Results. <http://www.pascalnetwork.org/challenges/VOC/voc2006/>.
- Fathy M. E., Hussein A. S., Tolba M. F.*, 2011. Fundamental matrix estimation: a study of error criteria. *Pattern Recognition Letters* 32 (2), 383–391.
- Faugeras O., Luong Q.-T., Maybank S. J.*, 1992. Camera self-calibration: theory and experiments. In: Sandini, G. (Ed.), Proc. 2nd European Conf. on Computer Vision. In: Lecture Notes in Computer Science, vol. 588. Springer-Verlag, Berlin Heidelberg, pp. 321–334.
- Feichtenhofer C., Pinz A., Zisserman A.*, 2017. Detect to track and track to detect. In: Proc. ICCV, pp. 3038–3046.
- Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D.*, 2010. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (9), 1627–1645.
- Fischler M. A., Bolles R. C.*, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24 (6), 381–395.
- Girshick R. B.*, 2015. Fast R-CNN. In: Proc. ICCV. 2015.
- Haralick R. M., Shanmugam K., Dinstein I.*, 1973. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics* 3 (6), 610–621.

- Harris C., Stephens M.*, 1988. A combined corner and edge detector. In: Proc. 4th Alvey Vision Conf., pp. 147–151.
- Hartley R. I.*, 1995. A linear method for reconstruction from lines and points. In: Proc. Int. Conf. on Computer Vision, pp. 882–887.
- Hartley R. I., Sturm P.*, 1994. Triangulation. In: American Image Understanding-Workshop, pp. 957–966.
- Hayman E., Caputo B., Fritz M., Eklundh J.-O.*, 2004. On the significance of real-world conditions for material classification. In: ECCV, pp. 253–266.
- He K., Zhang X., Ren S., Sun J.*, 2016. Deep residual learning for image recognition. In: Proc. CVPR.
- Held D., Thrun S., Savarese S.*, 2016. Learning to track at 100 fps with deep regression networks. In: Proc. ECCV. Springer, pp. 749–765.
- Hinton G. E.*, 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation* 14 (8), 1771–1800.
- Hinton G. E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. R.*, 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580v1 [cs.NE].
- Horprasert T., Harwood D., Davis L. S.*, 1999. A statistical approach for real-time robust background subtraction and shadow detection. In: Proc. IEEE ICCV Frame-Rate Applications Workshop, pp. 1–19.
- Hsiao J. Y., Sawchuk A. A.*, 1989a. Supervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (12), 1279–1292.
- Hsiao J. Y., Sawchuk A. A.*, 1989b. Unsupervised textured image segmentation using feature smoothing and probabilistic relaxation techniques. *Computer Vision, Graphics, and Image Processing* 48, 1–21.
- ImageNet, 2015. ImageNet large scale visual recognition challenge (ILSVRC2015). <http://image-net.org/challenges/LSVRC/2015/>.
- Janney P., Geers G.*, 2010. Texture classification using invariant features of local textures. *IET Image Processing* 4 (3), 158–171.
- Kaizer H.*, 1955. A quantification of textures on aerial photographs. MS thesis. Boston Univ.
- Kalal Z., Mikolajczyk K., Matas J.*, 2011. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (7), 1409–1422.
- Kanatani K.*, 1996. Statistical Optimization for Geometric Computation: Theory and Practice. Elsevier, Amsterdam, the Netherlands.
- Kitchen L., Rosenfeld A.*, 1982. Gray-level corner detection. *Pattern Recognition Letters* 1, 95–102.
- Krizhevsky A., Sutskever I., Hinton G. E.*, 2012. ImageNet classification with deep convolutional neural networks. In: Proc. 26th Annual Conf. on Neural Information Processing Systems. Lake Tahoe, Nevada, pp. 3–8.
- Laws K. I.*, 1979. Texture energy measures. In: Proc. Image Understanding Workshop, Nov., pp. 47–51.
- Laws K. I.*, 1980a. Rapid texture identification. In: Proc. SPIE Conf. on Image Processing for Missile Guidance, 238. San Diego, Calif. 28 July – 1 Aug., pp. 376–380.
- Laws K. I.*, 1980b. Textured Image Segmentation. PhD thesis. Univ. of Southern California, Los Angeles.

- LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D.*, 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1 (4), 541–551.
- LeCun Y., Bottou L., Bengio Y., Haffner P.*, 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- Lee S. H., Civera J.*, 2019. Triangulation: why optimize? vol. 23. arXiv:1907.11917v2 [cs.CV].
- Leibe B., Leonardis A., Schiele B.*, 2008. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision* 77 (1), 259–289.
- Lipton A. J., Fujiyoshi H., Patil R. S.*, 1998. Moving target classification and tracking from real-time video. In: *Proc. 4th IEEE Workshop on Applications of Computer Vision*, pp. 8–14.
- Liu L., Chen J., Fieguth P., Zhao G., Chellappa R., Pietikäinen M.*, 2019. From BoW to CNN: two decades of texture representation for texture classification. *International Journal of Computer Vision* 127, 74–109.
- Lo B. P. L., Velastin S. A.*, 2001. Automatic congestion detection system for underground platforms. In: *Proc. Int. Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 158–161.
- Longuet-Higgins H. C.*, 1981. A computer algorithm for reconstructing a scene from two projections. *Nature* 293, 133–135.
- Magee M. J., Aggarwal J. K.*, 1984. Determining vanishing points from perspective images. *Computer Vision, Graphics, and Image Processing* 26 (2), 256–267.
- Mathias M., Benenson R., Pedersoli M., Van Gool L.*, 2014. Face detection without bells and whistles. In: *Proc. 13th European Conf. on Computer Vision*. Zurich, Switzerland, 8–11 September.
- Mirmehdi M., Xie X., Suri J. (Eds.)*, 2008. *Handbook of Texture Analysis*. Imperial College Press, London.
- Nagel H.-H.*, 1983. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing* 21, 85–117.
- Noh H., Hong S., Han B.*, 2015. Learning deconvolution network for semantic segmentation. In: *Proc. IEEE Int. Conf. on Computer Vision*. Santiago, Chile, 13–16 December, pp. 1520–1528.
- Pietikäinen M., Rosenfeld A., Davis L. S.*, 1983. Experiments with texture classification using averages of local pattern matches. *IEEE Transactions on Systems, Man and Cybernetics* 13 (3), 421–426.
- Ren S., He K., Girshick R., Sun J.*, 2015. Faster R-CNN: towards real-time object detection with region proposal networks. arXiv:1506.01497 [cs.CV].
- Rosenfeld A., Troy E. B.*, 1970a. *Visual Texture Analysis*. Computer Science Center, Univ. of Maryland. Techn. Report TR-116.
- Rosenfeld A., Troy E. B.*, 1970b. Visual texture analysis. In: *Conf. Record for Symposium on Feature Extraction and Selection in Pattern Recognition*, IEEE Publication 70C-51C, Argonne, Ill., pp. 115–124.
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla, A. Bernstein M., Berg C., Fei-Fei L.*, 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (3), 211–252.

- Shah M. A., Jain R.*, 1984. Detecting time-varying corners. *Computer Vision, Graphics, and Image Processing* 28, 345–355.
- Simonyan K., Zisserman A.*, 2015. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556v6*.
- Stalder S., Grabner H., van Gool L.*, 2009. Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition. In: *IEEE 12Th Int. Conf. on Workshop on On-Line Learning for Computer Vision*.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.*, 2014. Going deeper with convolutions. *arXiv:1409.4842v1 [cs.CV]*.
- Tighe J., Lazebnik S.*, 2013. Finding things: image parsing with regions and per-exemplar detectors. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Portland, Oregon, 23–28 June, pp. 3001–3008.
- Torr P., Zisserman A.*, 1997. Performance characterization of fundamental matrix estimation under image degradation. *Machine Vision and Applications* 9 (5), 321–333.
- Usher M.*, 1986. Local linear transforms for texture measurements. *Signal Processing* 11, 61–79.
- Usher M., Eden M.*, 1989. Multiresolution feature extraction and selection for texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (7), 717–728.
- Usher M., Eden M.*, 1990. Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering. *IEEE Transactions on Systems, Man and Cybernetics* 20 (4), 804–815.
- Visnes R.*, 1989. Texture models and image measures for texture discrimination. *International Journal of Computer Vision* 3, 313–336.
- Wu Z., Thangali A., Sclaroff S., Betke M.*, 2012. Coupling detection and data association for multiple object tracking. In: *Proc. IEEE Conf. CVPR*, pp. 1948–1955.
- Yang Y., Li Z., Zhang L., Murphy C., Ver Hoeve J., Jiang H.*, 2012. Local label descriptor for example based semantic image labelling. In: *Proc. 12th European Conf. on Computer Vision*. Florence, Italy, 7–13 October, pp. 361–375.
- Zeiler M., Fergus R.*, 2014. Visualizing and understanding convolutional neural networks. In: *Proc. 13th European Conf. on Computer Vision*. Zurich, Switzerland, 8–11 September.
- Zhang L., Wu B., Nevatia R.*, 2007. Pedestrian detection in infrared images based on local shape features. In: *Proc. 3rd Joint IEEE Int. Workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*.
- Zuniga O. A., Haralick R. M.*, 1983. Corner detection using the facet model. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 30–37.

ОБ АВТОРЕ ГЛАВЫ

Рой Дэвис – почетный профессор факультета машинного зрения в Роял Холлоуэй, Лондонский университет. Он работал над многими аспектами зрения, от обнаружения признаков и подавления шума до робастного сопоставления

образов и реализации практических задач зрения в реальном времени. Область его интересов включает автоматизированный осмотр объектов, наблюдение, управление транспортными средствами и раскрытие преступлений. Он опубликовал более 200 статей и три книги: *Machine Vision: Theory, Algorithms, Practicalities* (1990 г.), *Electronics, Noise and Signal Recovery* (1993 г.) и *Image Processing for the Food Industry* (2000 г.); первая из них не теряет популярности на протяжении 25 лет, а в 2017 г. вышло ее значительно расширенное, пятое издание под названием *Computer Vision: Principles, Algorithms, Applications, Learning*. Рой является членом IoP и IET, а также старейшим членом IEEE. Он входит в редакционные коллегии журналов *Pattern Recognition Letters*, *Real-Time Image Processing*, *Imaging Science and IET Image Processing*. Получил степень доктора наук в Лондонском университете; в 2005 г. он был удостоен титула почетного члена BMVA, а в 2008 г. стал лауреатом премии Международной ассоциации распознавания образов.

Глава 2

Современные методы робастного обнаружения объектов

Авторы главы:

Чжаовэй Цай, Amazon Web Services,
Пасадена, Калифорния, США;

Нуно Васконселос, Калифорнийский университет в Сан-Диего,
факультет электроники и вычислительной техники,
Сан-Диего, Калифорния, США

Краткое содержание главы:

- знакомство с задачей обнаружения объектов в компьютерном зрении;
- рассмотрены некоторые усовершенствованные детекторы объектов, основанные на глубоких нейронных сетях, и некоторые методы, которые приобрели особое значение в публикациях по обнаружению объектов в последние годы.

2.1. ВВЕДЕНИЕ

Обнаружение объектов – одна из самых фундаментальных и сложных задач компьютерного зрения. Она обобщает более изученную задачу классификации объектов. При наличии конкретного изображения распознавание объектов ищет ответ на вопрос «что». Что за предметы изображены на рисунке? Например, изображение на рис. 2.1 включает в себя человека и судно. Помимо вопроса «что», обнаружение объектов также ищет ответ на вопрос «где». В каких областях изображения находится объект? Пример ответа показан на рис. 2.1b, где для разграничения области расположения каждого объекта используются ограничивающие прямоугольники.

Обнаружение объектов имеет множество практических применений. Например, автономные транспортные средства полагаются на обнаружение объектов для локализации объектов, понимания окружающей среды и при-

нения безопасных решений. В медицинской визуализации детекторы объектов могут помочь определить местонахождение поражений при медицинском сканировании, облегчая работу рентгенологов и других медицинских специалистов. Однако обнаружение объектов также часто является базовой операцией в компьютерном зрении, на результаты которой опираются многие последующие задачи, такие как визуальные ответы на вопросы, субтитры, визуальная навигация, захват предметов роботом, оценка позы и т. д. Например, обнаружение объекта может не только помочь роботу точно распознать объекты в физическом мире, но и позволяет ему понять семантику этого объекта – как он связан с другими объектами в сцене и какую роль он может играть в решении задачи, в одиночку или в команде роботов. Следовательно, развитие обнаружения объектов принесет пользу многим другим областям компьютерного зрения и сделает системы компьютерного зрения более эффективными в целом.

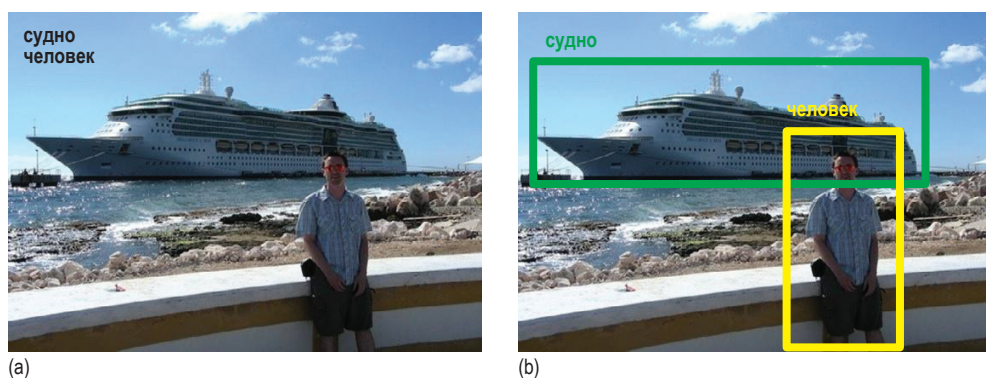


Рис. 2.1 ❖ Различие между классификацией и обнаружением объектов

Детектор объектов сталкивается со многими проблемами. Например, требуется точное обнаружение объектов нескольких категорий, масштабов, соотношений сторон и т. д., иногда в условиях плохого освещения, окклюзии и фоновых отвлекающих факторов. Это затрудняет разработку детекторов, достаточно надежных, чтобы хорошо работать в широком диапазоне реальных сценариев, что является необходимым условием для имитации зрительной системы человека.

У сложной проблемы обнаружения объектов существует долгая история исследований (Sung and Poggio, 1998; Rowley et al., 1996; Papageorgiou et al., 1998). Ранние работы были сосредоточены на обнаружении конкретных объектов, а именно лиц и людей, важных для многих приложений. Заметной вехой среди этих работ стал *детектор Виолы–Джонса* (VJ) (Viola and Jones, 2001, 2004). Это был первый детектор объектов в реальном времени для неограниченных сред, и он работал намного быстрее, чем все другие конкурирующие детекторы того времени. Идея состояла в том, чтобы сформулировать детектор как каскад классификаторов, которые поэтапно отвергают гипотезы об объекте, используя очень простые вейвлет-признаки Хаара (или прос-

то *признаки Хаара*). Добавляя дополнительные признаки на более поздних этапах, каскад может сформировать мощный детектор. Однако поскольку большинство гипотез можно отвергнуть с помощью простых признаков (на ранних стадиях), средняя вычислительная нагрузка невелика. Хотя вейвлеты работают быстро, они не очень точны. В более поздних работах была предложена *гистограмма ориентированных градиентов* (histogram of oriented gradients, HOG) (Dalal and Triggs, 2005) в качестве важного улучшения *функции масштабно-инвариантного преобразования признаков* (scale-invariant feature transform, SIFT) (Lowe, 1999, 2004). Метод HOG показал очень впечатляющую производительность, первоначально при обнаружении человека, а позже получил широкое распространение в различных задачах обнаружения объектов. Прорывом в обнаружении обобщенного объекта стало появление *модели деформируемых частей* (deformable part-based model, DPM) (Felzenszwalb et al., 2010). Она основана на признаках HOG, представляющих каждый объект как комбинацию корневой модели и деформируемых частей, где конфигурации фильтров частей были скрытыми переменными, изучаемыми автоматически. В 2007, 2008 и 2009 гг. модель DPM стала победителем конкурса Pascal VOC по обнаружению объектов (Everingham et al., 2010), который включает обнаружение 20 категорий объектов, таких как стол, автобус, человек или велосипед. Этот успех сделал DPM структурой по умолчанию для исследования обнаружения объектов до появления глубокого обучения.

В последние годы было показано, что представления изученных признаков, извлеченные с помощью глубоких сверточных нейронных сетей (CNN), значительно превосходят даже лучшие созданные вручную признаки, такие как SIFT, HOG и вейвлеты Хаара. Хотя обычные представления признаков CNN обладают высокой эффективностью классификации, их применение для обнаружения объектов требует нетривиальных расширений. В отличие от классификации, обнаружение объектов требует решения *двух* задач. Во-первых, детектор должен решить проблему распознавания, отличая объекты переднего плана от фона и присваивая им соответствующие метки классов объектов. Во-вторых, детектор должен решать проблему локализации, назначая точные ограничивающие рамки различным объектам. В этой главе мы рассмотрим системы обнаружения объектов на основе CNN, предложенные за последние несколько лет. Их можно разделить на две основные группы: *двухэтапные детекторы объектов*, такие как R-CNN (Girshick et al., 2014), SPP-Net (He et al., 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2017), MS-CNN (Cai et al., 2016), FPN (Lin et al., 2017a) и Cascade R-CNN (Cai, Vasconcelos, 2021), и *одноэтапные детекторы объектов*, включая YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) и RetinaNet (Lin et al., 2017).

2.2. ПРЕДВАРИТЕЛЬНЫЕ ПОЛОЖЕНИЯ

Большинство современных детекторов объектов реализуют комбинацию классификации и регрессии ограничивающей рамки. Классификация пытается предсказать класс объекта в области изображения, а регрессия огра-

ничающей рамки пытается определить область расположения объекта, предсказывая самую узкую рамку, содержащую объект. Рассмотрим эталон ограничительной рамки \mathbf{g} , связанный с меткой класса y , и гипотезу обнаружения \mathbf{x} ограничительной рамки \mathbf{b} . Поскольку \mathbf{b} обычно включает в себя объект и некоторое количество фона, может быть трудно определить, правильно ли обнаружен объект. Обычно это решается с помощью метрики *пересечения по объединению* (intersection over union, IoU):

$$\text{IoU}(\mathbf{b}, \mathbf{g}) = \frac{\mathbf{b} \cap \mathbf{g}}{\mathbf{b} \cup \mathbf{g}}. \quad (2.1)$$

Если IoU выше порога u , \mathbf{x} считается примером класса объекта ограничивающей рамки \mathbf{g} и обозначается как «положительный» пример. Таким образом, метка класса гипотезы \mathbf{x} является функцией u :

$$y_u = \begin{cases} y, & \text{IoU}(\mathbf{b}, \mathbf{g}) \geq u \\ 0, & \text{в остальных случаях} \end{cases}. \quad (2.2)$$

Если IoU не превышает порога для любого объекта, \mathbf{x} считается фоном и обозначается как «отрицательный» пример.

Хотя нет необходимости определять положительные/отрицательные примеры для задачи регрессии ограничивающей рамки, порог IoU u также требуется для выбора набора образцов

$$\mathcal{G} = \{(\mathbf{g}_i, \mathbf{b}_i) | \text{IoU}(\mathbf{g}_i, \mathbf{b}_i) \geq u\}, \quad (2.3)$$

используемого для обучения регрессора. Хотя пороговые значения IoU, применяемые для двух задач, не обязательно должны быть идентичными, на практике это обычное дело. Следовательно, порог IoU u определяет качество детектора. Большие пороги способствуют тому, чтобы обнаруженные ограничивающие рамки были точно совмещены с эталонами. Небольшие пороги вознаграждают детекторы, которые создают свободные ограничивающие рамки с небольшим перекрытием с эталоном. Некоторые примеры гипотез повышения качества показаны на рис. 2.2.

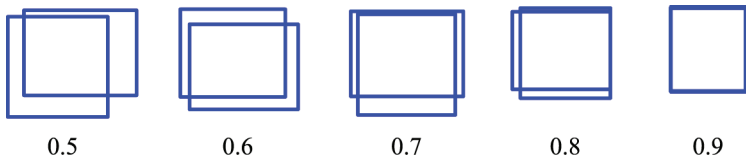


Рис. 2.2 ❖ Примеры повышения качества. Числа представляют собой значения IoU (2.1) между двумя ограничивающими прямоугольниками, указывая, насколько хорошо они совмещены друг с другом

2.3. R-CNN

Сеть R-CNN (Girshick et al., 2014) (Regions with CNN, области с CNN) была новаторской попыткой использовать глубокие нейронные сети для обобщенного обнаружения объектов. Это была первая работа, которая превзошла методы в стиле DPM за счет использования мощных представлений признаков CNN. Она также продемонстрировала, что признаки CNN, предварительно обученные для классификации в ImageNet (Russakovsky et al., 2015), могут быть успешно уточнены для других последующих задач, например обнаружения, сегментации и т. д.

2.3.1. Внутреннее устройство

R-CNN состоит из трех модулей, показанных на рис. 2.3. Поскольку вычисления CNN являются дорогостоящими, первым шагом является создание предложений областей, не зависящих от категорий, с использованием выборочного поиска (van de Sande et al., 2011). Эти предложения определяют набор обнаружений-кандидатов, доступных для детектора, уменьшая количество гипотез обнаружения с миллионов до тысяч. Второй шаг – извлечение признаков из каждой предложенной области с использованием CNN, обученной распознаванию, например AlexNet (Krizhevsky et al., 2012) или VGG-Net (Simonyan, Zisserman, 2014). Обнаруженные предложения произвольного масштаба и размеров сначала обрезаются и деформируются до размера входных данных сети. Затем изображения с измененным размером пропускаются через CNN, а выходные данные предпоследнего сетевого слоя используются в качестве представления признаков для каждого предложения. Наконец, третий модуль, реализованный с помощью линейных SVM для конкретных классов, создает прогнозы классов для предложений. Для лучшей локализации применяются дополнительные регрессоры ограничивающей рамки для уточнения ограничивающих рамок обнаруженных объектов.

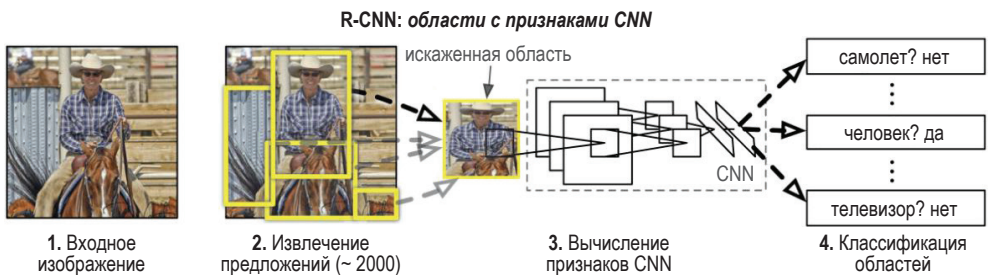


Рис. 2.3 ❖ Внутреннее устройство R-CNN

2.3.2. Обучение

CNN, используемая для извлечения признаков, предварительно обучена в задаче классификации ImageNet (Russakovsky et al., 2015) и точно настроена на искаженные области предложений, используемые в задаче обнаружения. Точная настройка представляет собой задачу классификации $K + 1$ с K категориями объектов и одним фоновым классом (например, $K = 20$ для набора данных VOC (Everingham et al., 2010) и $K = 80$ для набора данных COCO (Lin et al., 2014)). Поскольку CNN нуждаются в больших объемах данных, недостаточно использовать эталоны только в качестве положительных примеров. Решение состоит в том, чтобы использовать все предложения с IoU выше 0,5 относительно ближайшей рамки эталона как положительные, а остальные – как отрицательные. Во время обучения важно соблюдать сбалансированное соотношение между положительными и отрицательными предложениями. На практике положительные и отрицательные результаты отбираются равномерно из пула эталонов с соотношением 1:3 в каждой обучающей партии. Тонкая настройка сводит к минимуму перекрестную энтропийную потерю

$$L_{cls}(h(\mathbf{x}), y) = -\log h_y(\mathbf{x}), \quad (2.4)$$

где \mathbf{x} – предложение по классификации, y – метка класса, а $h(\mathbf{x})$ – классификатор. После тонкой настройки линейные классификаторы SVM для конкретных категорий и регрессоры ограничительной рамки обучаются на признаках предложения, сгенерированных CNN. Эта многоступенчатая процедура обучения извлечения признаков, точной настройки CNN с кросс-энтропийной потерей, обучения SVM и настройки регрессоров с ограничительной рамкой является медленной, утомительной и неэлегантной.

2.4. Сеть SPP-Net

Хотя R-CNN значительно повысил общую производительность обнаружения объектов, это сложный детектор, поскольку дорогостоящие вычисления CNN повторяются для тысяч предложений, полученных из каждого отдельного изображения. В результате прогон детектора R-CNN на каждом изображении может занять более 30 с. Поскольку предложения, извлеченные из изображения, имеют наибольшее количество пикселей, большинство этих вычислений являются избыточными. Эту избыточность удалось уменьшить с помощью SPP-Net (He et al., 2014), которая разделяла вычисления между предложениями.

В отличие от конвейера R-CNN, который обрезает предложения перед вычислением CNN, как показано на рис. 2.4 (вверху), SPP-Net пересылает изображение через сверточные слои сети целиком. Затем используется пулинг пространственных пирамид (spatial pyramid pooling, Lazebnik et al., 2006) для извлечения признаков фиксированной длины из обрезанных карт признаков, связанных с каждым предложением. Эти признаки фикси-

рованной длины, наконец, вводятся в набор полностью связанных слоев для окончательного прогноза, как показано на рис. 2.4 (внизу). Это простое изменение позволяет совместно использовать дорогостоящие вычисления CNN между предложениями, и все изображение обрабатывается только один раз. Важной операцией является *пространственный пирамидальный пулинг* (spatial pyramid pooling, SPP), операция, показанная на рис. 2.5, которая отображает экзemplярные признаки произвольного масштаба и размера в вектор фиксированной длины. Это было первое доказательство того, что для признаков из сверточной карты признаков можно выполнить пространственный пулинг для создания экзemplярного представления объектов с хорошими свойствами для распознавания экзemplяров. Это вдохновило исследователей на последующие разработки, такие как Fast R-CNN.



Рис. 2.4 ❖ Сравнение конвейеров R-CNN (вверху) и SPP-Net (внизу)

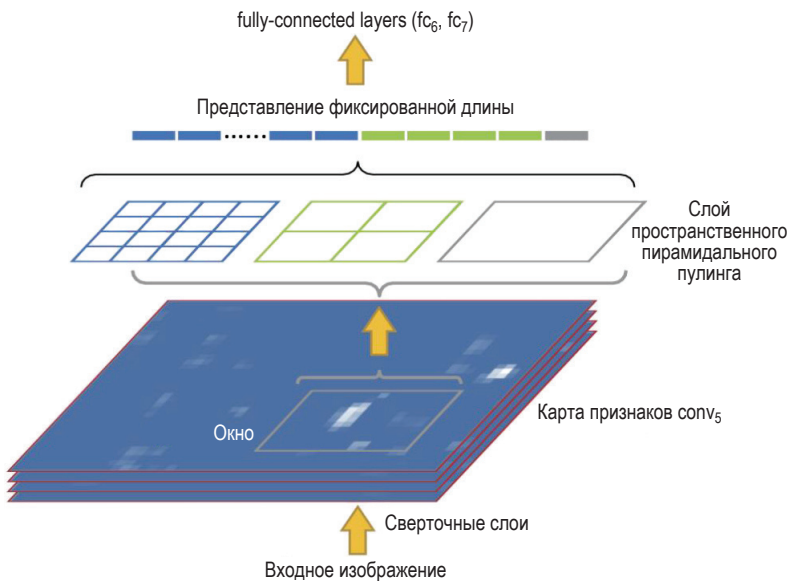


Рис. 2.5 ❖ Конвейер SPP-Net для обнаружения объектов

2.5. СЕТЬ FAST R-CNN

Архитектура SPP-Net наследует утомительную многоступенчатую процедуру обучения R-CNN. В архитектуре Fast R-CNN это делается значительно проще. В этом подходе извлечение признаков, точное дообучение сети для новой

задачи, классификация по экземплярам и регрессия с ограничительной рамкой интегрированы в единую структуру, что позволяет легко использовать обнаружение объектов на основе глубокого обучения.

2.5.1. Архитектура

Конвейер Fast R-CNN показан на рис. 2.6. Подобно SPP-Net, сеть Fast R-CNN пересылает все изображение через сверточные слои CNN для создания карт признаков. Затем слой пулинга *видимой области* (region of interest, RoI) применяется для извлечения вектора признаков фиксированной длины для каждого предложения объекта. Наконец, два полносвязных (fully connected, FC) слоя используются для окончательных прогнозов: вероятности классификации для $K + 1$ классов и регрессии четырех координат ограничивающей рамки. В отличие от R-CNN и SPP-Net, Fast R-CNN обучается от начала до конца с многозадачной функцией потерь, что позволяет избежать утомительной многоэтапной процедуры обучения.

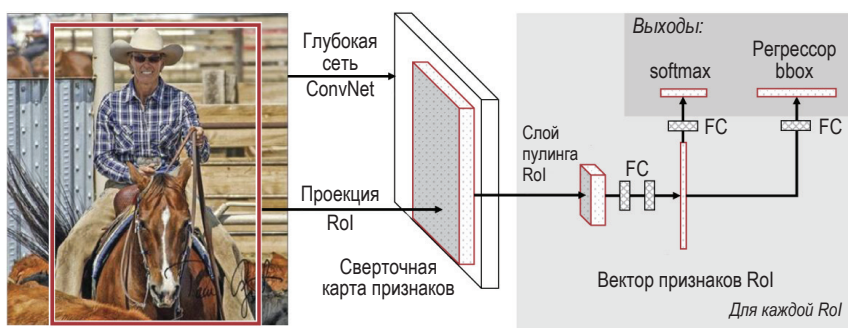


Рис. 2.6 ❖ Конвейер Fast R-CNN

2.5.2. Пулинг ROI

Операция пулинга RoI представляет собой более простую версию пространственного пирамидального пулинга, показанного на рис. 2.5. Вместо того чтобы выполнять пулинг RoI с высотой и шириной (h, w) в различные пространственные разрешения ($1 \times 1, 2 \times 2$ и 4×4) и объединять их вместе, как в SPP, при объединении RoI используется одно разрешение $H \times W$, например 7×7 . При заданном окне RoI размером $h \times w$ пулинг RoI делит его на подокна $H \times W$, каждое из которых имеет размер $h/H \times w/W$. Затем внутри каждого подокна применяется max-пулинг для извлечения наибольшего значения признака. Этот процесс применяется независимо к каждому каналу карты признаков. Хотя пулинг RoI проще, чем SPM, он все же может эффективно извлекать мощное представление признаков для предложений произвольного размера и масштаба из предварительно вычисленных сверточных карт признаков. Это критическое требование для функции обнаружения объекта.

2.5.3. Многозадачная функция потерь

Fast R-CNN обучается двум задачам обучения: классификации и регрессии ограничивающей рамки. Они совместно оптимизируются во время обучения с использованием многозадачной функции потерь

$$L = L_{cls}(h(\mathbf{x}), y) + \lambda[y \geq 1]L_{loc}(f(\mathbf{x}, \mathbf{b}), \mathbf{g}), \quad (2.5)$$

где λ управляет балансом между двумя требованиями. $[y \geq 1]$ равно 1, когда $y \geq 1$, и равно 0 в противном случае, что означает отсутствие регрессионной потери ограничивающей рамки для фонового класса.

Классификация

Классификатор представляет собой функцию $h(\mathbf{x})$, которая присваивает участок изображения \mathbf{x} одному из $K + 1$ классов, где класс 0 содержит фон, а остальные классы – объекты для обнаружения. Фактически $h(\mathbf{x})$ – это $(K + 1)$ -мерная оценка апостериорного распределения по классам, т. е. $h_k(\mathbf{x}) = p(y = k|\mathbf{x})$, где y – метка класса. L_{cls} – кросс-энтропийные потери (2.4).

Регрессия ограничивающей рамки

Ограничивающая рамка $\mathbf{b} = (b_x, b_y, b_w, b_h)$ содержит четыре координаты фрагмента изображения \mathbf{x} . Регрессия ограничивающей рамки направлена на регрессию потенциальной ограничивающей рамки \mathbf{b} в целевую ограничивающую рамку \mathbf{g} с использованием регрессора $f(\mathbf{x}, \mathbf{b})$. Регрессионная функция потерь имеет вид

$$L_{loc}(a, \mathbf{b}) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(a_i - b_i), \quad (2.6)$$

где

$$\text{smooth}_{L_1}(x) = \begin{cases} 0,5x^2, & |x| < 1 \\ |x| - 0,5 & \text{в остальных случаях} \end{cases} \quad (2.7)$$

– это гладкая функция потерь L_1 . Это комбинация потерь L_1 и L_2 , которая ведет себя как потеря L_1 , когда $|x| < 1$, и потеря L_2 в противном случае. Она исправляет негладкое поведение потери L_1 , т. е. когда градиент равен -1 , при отрицательном x , и 1 в противном случае. Плавная потеря L_1 может обеспечить более стабильное поведение при обучении.

Для поддержания инвариантности к масштабу и местоположению smooth_{L_1} работает с вектором расстояния $\Delta = (\delta_x, \delta_y, \delta_w, \delta_h)$, определяемым уравнениями

$$\begin{aligned} \delta_x &= \frac{g_x - b_x}{b_w}, & \delta_y &= \frac{g_y - b_y}{b_h}, \\ \delta_w &= \log(g_w/b_w), & \delta_h &= \log(g_h/b_h). \end{aligned} \quad (2.8)$$

Поскольку регрессия ограничивающей рамки обычно выполняет незначительные корректировки \mathbf{b} , численные значения (2.8) могут быть очень малы. Это обычно делает потери регрессии намного меньшими, чем потери классификации. Для повышения эффективности многозадачного обучения Δ нормализуется по среднему значению и дисперсии, например δ_x заменяется на

$$\delta'_x = \frac{\delta_x - \mu_x}{\sigma_x}. \quad (2.9)$$

2.5.4. Стратегия тонкой настройки

Выборка

R-CNN и SPP-Net берут выборку RoI. Это может привести к очень неэффективному обучению, поскольку видимые области извлекаются из разных изображений и каждое изображение требует полного прямого вычисления CNN. Чтобы избежать этой проблемы, Fast R-CNN сначала делает выборку N изображений, из которых затем выбирает R RoI для каждого изображения. Выбирая $N \ll R$, можно ограничиться вычислением CNN только для небольшого количества (N) изображений. Однако возникает опасение, что выбранные видимые области коррелируют, и это может замедлить конвергенцию обучения. Однако на практике данная стратегия доказала свою эффективность (Girshick, 2015; Ren et al., 2017). RoI отбираются из каждого изображения, чтобы получить 25 % положительных и 75 % отрицательных обучающих примеров.

Обратное распространение через пулинг RoI

Другим важным улучшением Fast R-CNN по сравнению с сетью SPP было обратное распространение градиента через слой пулинга RoI. В отсутствие этого сверточные слои ниже слоя пулинга RoI не будут точно настроены на задачу обнаружения, как в случае с сетью SPP. Поскольку в пуле RoI каждый выходной объект является результатом max-пулинга соответствующего подокна на карте признаков, вычисления обратного распространения сводятся к операциям max-пулинга. А именно обратное распространение выходного градиента происходит только до положения наибольшего максимального значения признака в подокне. Эта стратегия применяется к каждому признаку RoI каждого предложения области.

2.6. FASTER R-CNN

Архитектуры SPP-Net и Fast R-CNN значительно улучшили скорость работы R-CNN приблизительно с 30 до 2 секунд на изображение. Они выполняли общий этап обнаружения предложений, который опирался на низкоуровневые признаки, такие как пиксели и края, и работали на CPU, который обладает ограниченной скоростью вычислений. Например, выборочный детектор

предложений требует около 2 секунд на изображение. Архитектура Faster R-CNN устранила эту проблему за счет *сети прогнозирования регионов* (region proposal network, RPN), которая использует графические процессоры и общие вычисления признаков с сетью Fast R-CNN.

2.6.1. Архитектура

Как показано на рис. 2.7, Faster R-CNN состоит из двух модулей: сети прогнозирования регионов (RPN), которая предлагает регионы и является полностью сверточной, и детектора Fast R-CNN, который классифицирует эти предложения. В отличие от архитектур R-CNN и Fast R-CNN, вся система представляет собой единую, унифицированную и сквозную сеть для обнаружения объектов. Поскольку RPN разделяет большую часть своих вычислений с сетью Fast R-CNN, сама по себе RPN добавляет немного вычислительных затрат. Это позволяет Faster R-CNN сократить время генерации предложений и работать в реальном времени на современном графическом процессоре.

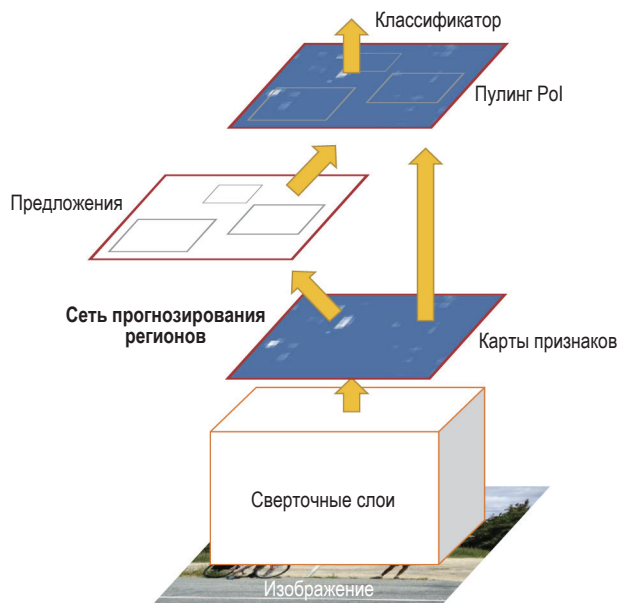


Рис. 2.7 ❖ Архитектура Faster R-CNN

2.6.2. Сети прогнозирования регионов

Предложения регионов обнаруживаются путем скольжения небольшой сети по сверточной карте признаков, как показано на рис. 2.8. Эта небольшая сеть реализована с помощью 256-мерного сверточного слоя 3×3 , слоя ReLU и двух полностью связанных выходных слоев. Подобно окончательным выходным слоям Fast R-CNN, первый выходной слой предназначен для бинарной клас-

сификации (передний план / фон), а второй – для регрессии ограничивающей рамки. Это дает *показатель объектности* (objectness score) и 4 координаты для данной привязки. В соответствии с показателем объектности RPN генерирует 300 лучших предложений, которые будут использоваться на более позднем этапе Fast R-CNN.

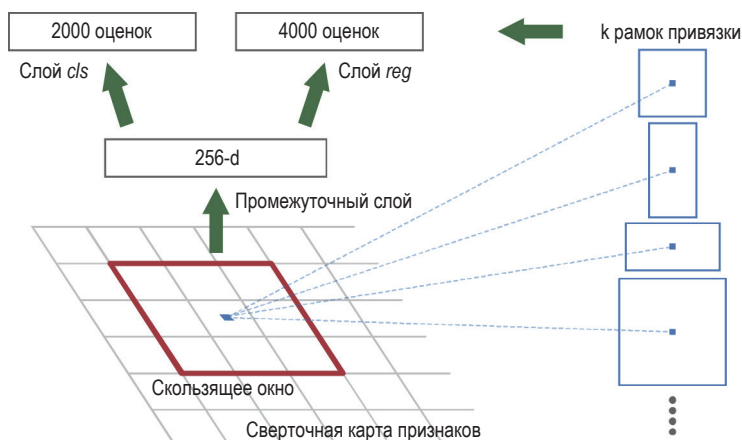


Рис. 2.8 ❖ Иллюстрация сети прогнозирования регионов

Привязки

Каждое местоположение скользящего окна должно, в принципе, генерировать одно предложение, поскольку каждое местоположение на карте объектов соответствует одному местоположению на входном изображении. Однако RPN одновременно прогнозирует k предложений регионов для каждого местоположения скользящего окна, чтобы учесть различные размеры объектов и соотношения сторон. Это становится возможно благодаря концепции *привязок* (anchor). В каждом конкретном месте предложение связано с *привязкой*, которая центрируется в центре скользящего окна и имеет собственный масштаб и соотношение сторон. Обычной практикой является использование $k = 9$ привязок, трех разных масштабов и трех разных соотношений сторон, для каждого положения скользящего окна. Каждая привязка создает шестимерное предложение, где четыре измерения кодируют координаты для регрессии ограничивающей рамки, а оставшиеся два – вероятности классов переднего плана и фона.

Обучение

Функция потерь RPN такая же, как (2.5). Ограничивающие рамки привязки необходимы для регрессии к соответствующим эталонным рамкам. Для балансировки привязки выбираются во время обучения таким образом, чтобы соотношение между положительными и отрицательными привязками составляло 1:1. Обратите внимание, что соотношение здесь отличается от соотношения 1:3 при обучении Fast R-CNN, упомянутом в разделе 2.5, потому что задача RPN состоит в том, чтобы обнаружить как можно больше предложений.

При более высокой доле положительных привязок модель будет поощряться к обнаружению большего количества положительных результатов. Если брать 300 лучших предложений, сгенерированных RPN, обучение детектора Fast R-CNN остается таким же, как указано выше. Вычисления сверточных признаков совместно используются RPN и Fast R-CNN, и вся сеть может быть обучена от начала до конца с помощью стандартного обратного распространения и *стохастического градиентного спуска* (stochastic gradient descent, SGD).

2.7. Каскадная R-CNN

Проблема обнаружения сложна, отчасти из-за того, что существует много «близких» ложных срабатываний, соответствующих «близким, но неправильным» ограничивающим рамкам. Эффективный детектор должен обнаруживать все истинные срабатывания на изображении, подавляя при этом близкие ложные срабатывания. Качество гипотезы обнаружения определяется ее IoU с эталоном, а качество детектора – порогом IoU, используемым для его обучения.

Высококачественное обнаружение

Проблема заключается в том, что независимо от выбора порога IoU и настройка обнаружения является крайне противоречивой. Когда значение u высокое, позитивные предложения содержат мало фона, но трудно собрать большие позитивные обучающие наборы. Когда значение u низкое, возможны более богатые и разнообразные положительные обучающие наборы, но у обученного детектора меньше стимулов отбрасывать близкие ложные срабатывания. В общем, очень сложно добиться того, чтобы один классификатор одинаково хорошо работал на всех уровнях IoU. Кроме того, поскольку большинство гипотез, выдаваемых детекторами предложений (такими как RPN или выборочный поиск), имеют низкое качество, детектор объектов наверху сети должен различать гипотезы более низкого качества. Стандартным компромиссом между этими противоречивыми требованиями является выбор значения $u = 0,5$, которое используется почти во всех современных детекторах объектов. Это, однако, относительно низкий порог, что затрудняет обучение детекторов, которые могут эффективно отклонять близкие ложные срабатывания.

Как правило, детектор достигает высокого качества только в том случае, если ему представлены высококачественные предложения. Этого, однако, нельзя добиться, просто увеличивая порог u во время обучения. Наоборот, установка высокого значения u обычно ухудшает качество обнаружения. Эта проблема, то есть тот факт, что обучение детектора с более высоким порогом приводит к снижению качества, называется *парадоксом качественного обнаружения* (paradox of high-quality detection). У него две причины. Во-первых, механизмы предложения объектов склонны создавать распределения гипотез, сильно смещенные в сторону низкого качества. В результате использование больших порогов IoU во время обучения экспоненциально уменьшает

количество положительных обучающих примеров. Это особенно проблематично для нейронных сетей, которые интенсивно используют примеры, что делает стратегию обучения с «высоким u » очень склонной к переобучению. Во-вторых, существует несоответствие между качеством детектора и качеством гипотез, доступных во время вывода. Поскольку детекторы высокого качества оптимальны только для гипотез высокого качества, качество обнаружения может существенно ухудшиться для гипотез более низкого качества. Каскадная архитектура R-CNN решает эту проблему, позволяя использовать детекторы объектов высокого качества.

2.7.1. Каскадная архитектура R-CNN

Каскадная R-CNN (Cai, Vasconcelos, 2021) представляет собой многоступенчатое расширение Faster R-CNN, как показано на рис. 2.9. Вместо одного детектора она использует каскад детекторов, которые последовательно более избирательны в отношении ложных срабатываний на близкие предложения. Пороги IoU обычно составляют 0,5, 0,6 и 0,7 для различных ступеней обнаружения. Каскад ступеней R-CNN обучается последовательно, используя выходные данные одной ступени для обучения следующей. Этот метод основан на наблюдении, что выходной IoU регрессора ограничивающей рамки почти всегда лучше, чем его входной IoU. В результате выходные данные детектора, обученного определенному порогу IoU, являются хорошим начальным распределением гипотез для обучения следующего детектора более высокому порогу IoU. Настраивая ограничивающие рамки, каждый этап стремится найти хороший набор близких ложных срабатываний для обучения следующей ступени. Основным результатом этой повторной выборки является постепенное повышение качества гипотез обнаружения от одной ступени к другой. В результате последовательность детекторов решает две проблемы, лежащие в основе парадокса высококачественного обнаружения. Во-первых, поскольку операция повторной выборки гарантирует наличие большого количества примеров для обучения всех детекторов в последовательности, можно обучать детекторы с высоким IoU без переобучения. Во-вторых, использование одной и той же каскадной процедуры во время вывода дает набор гипотез все более высокого качества, хорошо согласующихся с воз-

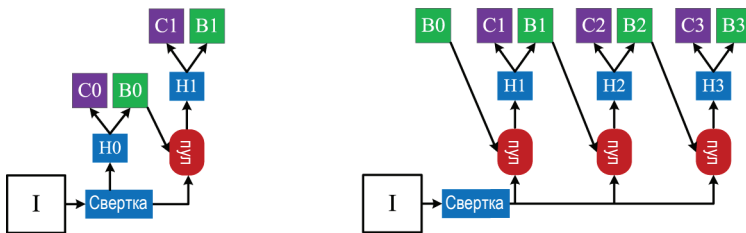


Рис. 2.9 ❖ Каскадная R-CNN – это многоступенчатое расширение Faster R-CNN. Здесь «I» – это входное изображение, «свертка» – предшествующие свертки, «лп» – извлечение признаков по регионам, «H» – ветвь (head) сети, «B» – ограничивающая рамка, «C» – классификация, «B0» – предложения во всех архитектурах

растающим качеством каскадов детектора. Это обеспечивает более высокую точность обнаружения.

2.7.2. Каскадная регрессия ограничивающей рамки

Поскольку одному регрессору сложно одинаково хорошо работать на всех уровнях качества, в каскадной R-CNN задача регрессии разбивается на последовательность более простых шагов. Она состоит из каскада специализированных регрессоров

$$f(\mathbf{x}, \mathbf{b}) = f_T \circ f_{T-1} \circ \dots \circ f_1(\mathbf{x}, \mathbf{b}), \quad (2.10)$$

где T – общее количество ступеней каскада. Ключевым моментом является то, что каждый регрессор f_t оптимизирован для распределения ограничивающей рамки $\{\mathbf{b}^t\}$, сгенерированного предыдущим регрессором, а не для начального распределения $\{\mathbf{b}^1\}$. Таким образом, гипотезы постепенно улучшаются (этот эффект называется *прогрессивным улучшением*). Эффективность каскадной регрессии иллюстрирует рис. 2.10, на котором изображено распределение вектора расстояния регрессии $\Delta = (\delta_x, \delta_y, \delta_w, \delta_h)$ из (2.8) на разных ступенях каскада. Обратите внимание, что большинство гипотез становятся ближе к истине по мере продвижения по каскаду.

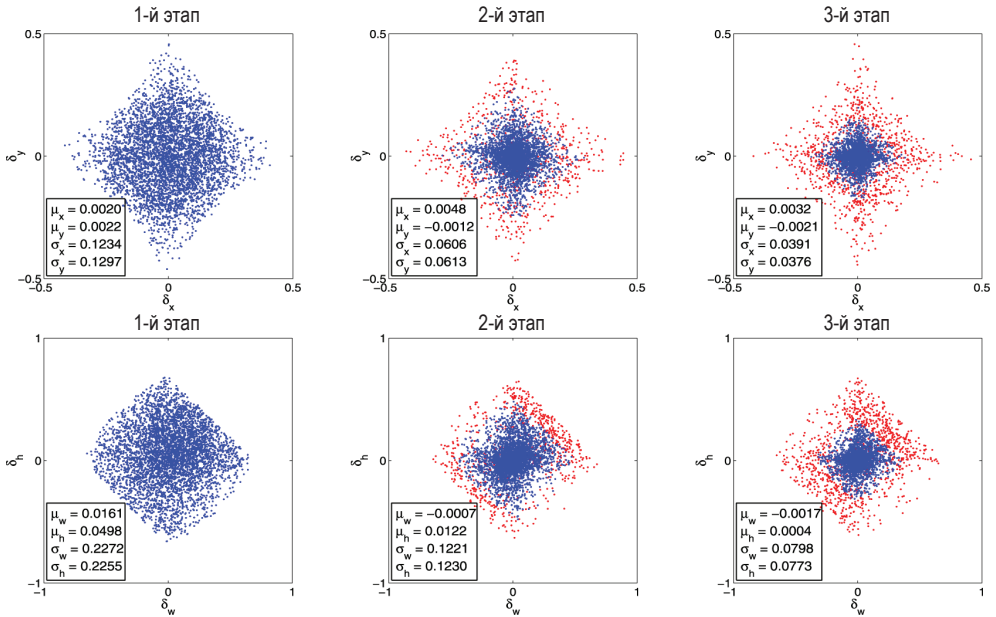


Рис. 2.10 ❖ Распределение вектора расстояния Δ из (2.8) (без нормировки) на разных ступенях каскада. Вверху: график (δ_x, δ_y) . Внизу: график (δ_w, δ_h) . Красные точки – это выбросы для увеличения порогов IoU на более поздних этапах, а показанная статистика получена после удаления выбросов

2.7.3. Каскадное обнаружение

Высококачественный детектор трудно обучить напрямую. Каскадная R-CNN решает проблему, используя каскадную регрессию в качестве механизма повторной выборки. Каскадная регрессия начинается с образцов $(\mathbf{x}_i, \mathbf{b}_i)$ и используется для последовательной повторной выборки распределения образцов $(\mathbf{x}'_i, \mathbf{b}'_i)$ с более высоким IoU. Это позволяет наборам положительных образцов последовательных ступеней сохранять примерно постоянный размер по мере увеличения качества детектора u .

На каждом этапе t ветка R-CNN включает классификатор h_t и регрессор f_t , оптимизированный для соответствующего порога IoU u^t , где $u^t > u^{t-1}$. Они обучаются с функцией потерь

$$L(\mathbf{x}^t, g) = L_{cls}(h_t(\mathbf{x}^t), y^t) + \lambda[y^t \geq 1]L_{loc}(f_t(\mathbf{x}^t, \mathbf{b}^t), \mathbf{g}), \quad (2.11)$$

где $\mathbf{b}^t = f_{t-1}(\mathbf{x}^{t-1}, \mathbf{b}^{t-1})$, \mathbf{g} – эталонный объект для \mathbf{x}^t , $\lambda = 1$ – коэффициент компромисса, y^t – метка \mathbf{x}^t по критерию u^t согласно (2.2) и $[\cdot]$ – индикаторная функция. Обратите внимание, что использование $[\cdot]$ подразумевает, что порог IoU и регрессии ограничивающей рамки идентичен тому, который используется для классификации. Это каскадное обучение имеет два важных следствия для обучения детекторов. Во-первых, уменьшается возможность переобучения при больших порогах IoU u , поскольку положительных примеров становится много на всех этапах. Во-вторых, детекторы более глубоких ступеней оптимальны для более высоких порогов IoU. Это одновременное улучшение гипотез и качества детектора позволяет каскадной R-CNN преодолеть парадокс высокого качества обнаружения. При выводе применяется тот же каскад. Качество гипотез улучшается последовательно, и более качественные детекторы требуются только для работы с более качественными гипотезами, для которых они оптимальны.

2.8. ПРЕДСТАВЛЕНИЕ РАЗНОМАСШТАБНЫХ ПРИЗНАКОВ

Распознавание объектов, представленных на изображении в разном масштабе, является фундаментальной проблемой компьютерного зрения. Классическое решение в литературе состоит в том, чтобы полагаться на *пирамиды изображений*, такие как те, что показаны на рис. 2.11а, где исходное изображение последовательно масштабируют до нескольких изображений разного размера, из которых извлекают признаки (Viola, Jones, 2004; Felzenszwalb et al., 2010; Доллар и др., 2014). Применяя детектор с фиксированным масштабом ко всем изображениям элементов пирамиды, можно обнаруживать объекты в разных масштабах без потери точности. Маленькие (большие) объекты обнаруживаются в каналах большого (малого) разрешения пирамиды признаков. Тем не менее построение пирамиды признаков CNN требует больших вычислительных ресурсов, что делает это решение непрактичным

для большинства реальных приложений. Разработка эффективных представлений признаков CNN для различных масштабов является важным направлением исследований в области обнаружения объектов.

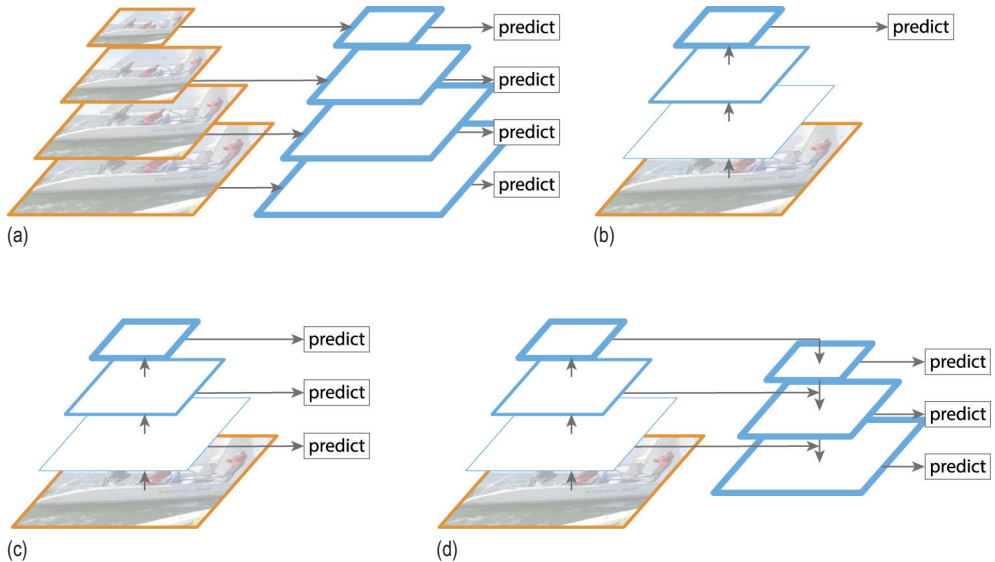


Рис. 2.11 ❖ (a) Пирамида изображения: признаки вычисляются для каждого масштаба изображения независимо. (b) Карта объектов с одним масштабом: обнаружение объектов работает только на карте объектов с одним масштабом в CNN. (c) Пирамида признаков: пирамида признаков для многомасштабного обнаружения, но с единой шкалой ввода изображения. (d) Сеть функциональных пирамид (FPN): FPN добавляет нисходящие соединения к пирамиде функций из (c), обеспечивая более инвариантное к масштабу семантическое представление функций в разных масштабах

Несмотря на большой успех детекторов объектов на основе глубокого обучения (Girshick et al., 2014; Girshick, 2015; He et al., 2014; Ren et al., 2017), в обнаружении объектов разного масштаба пока не удалось добиться хороших результатов. Как было сказано выше, R-CNN, SPP-Net и Fast R-CNN отбирают предложения объектов в нескольких масштабах, используя этап предварительного внимания, например выборочный поиск (van de Sande et al., 2011), а затем преобразуют эти предложения в фиксированный размер, поддерживаемый CNN. Но это лишь отодвигает проблему инвариантности масштаба на стадию внимания, которая не обучается совместно с CNN. Faster R-CNN (Ren et al., 2017) решает проблему совместного обучения, используя RPN для создания предложений нескольких масштабов. Однако, как показано на рис. 2.11b, это делается путем сдвига фиксированного набора фильтров по одному набору сверточных карт признаков. Это создает несоответствие между объектами переменного размера и фильтрами фиксированного рецептивного поля. Как показано на рис. 2.12, фиксированное рецептивное поле не может охватывать множество масштабов, в которых объекты пред-

стают в естественных сценах. В результате снижается эффективность обнаружения, особенно для небольших объектов, подобных показанному в центре рис. 2.12, которые довольно трудно обнаружить. Было предложено несколько сетей, расширяющих архитектуру двухэтапного детектора путем введения многомасштабных расширений RPN.

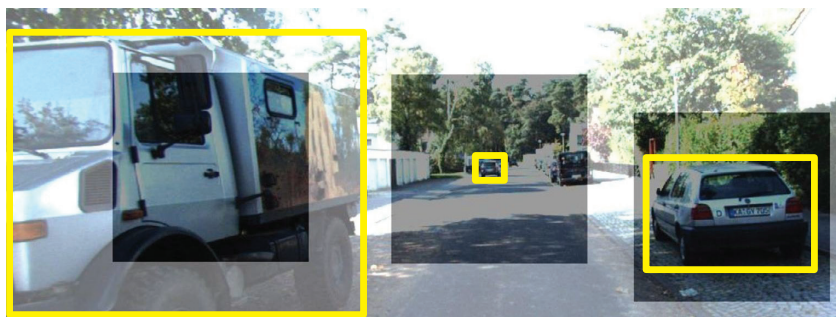


Рис. 2.12 ❖ На естественных изображениях объекты могут появляться в очень разных масштабах, что показано желтыми ограничивающими прямоугольниками. Один фильтр рецептивного поля фиксированного размера (показано в заштрихованной области) не может охватить весь разброс масштабов

2.8.1. Архитектура MC-CNN

Архитектура MS-CNN была предложена специально для решения проблемы многомасштабного обнаружения объектов. Она использует стратегию, альтернативную дорогостоящему вычислению пирамид изображений, опираясь на тот факт, что глубокие нейронные сети уже вычисляют иерархию признаков слой за слоем. Учитывая, что слои более высокого уровня подвергаются субдискретизации, эта иерархия даже имеет многоуровневую пирамидальную структуру. Следовательно, несоответствие между размерами объектов и рецептивных полей может быть устранено путем простого добавления выходных слоев на нескольких этапах сети, как показано на рис. 2.11с. Таким образом, MS-CNN реализует несколько детекторов, которые специализируются на различных диапазонах размеров. В то время как детекторы, основанные на более низких сетевых уровнях, таких как «conv-3», имеют меньшие рецептивные поля и лучше подходят для обнаружения небольших объектов, детекторы, основанные на более высоких уровнях, таких как «conv-5», лучше всего подходят для обнаружения крупных объектов. Комплементарные детекторы на выходах разных слоев объединяются в сильный *многомасштабный детектор*.

2.8.1.1. Архитектура

Подробная архитектура сети предложений MS-CNN показана на рис. 2.13. Сеть обнаруживает объекты, пропуская изображения через несколько ветвей обнаружения, которые объединяются в окончательный набор предложений. Она имеет стандартный ствол CNN, изображенный в центре рисунка, и набор

ми представлениями. В то время как карты с высоким разрешением содержат информацию о признаках семантики низкого уровня, таких как края, углы и т. д., карты с низким разрешением передают семантически богатую информацию, такую как категория объекта. Следовательно, карты высокого разрешения в основном информируют о местоположении объекта, а карты низкого разрешения – о его идентичности. Архитектура на рис. 2.11(с) может иметь неоптимальную производительность обнаружения, поскольку запрашивает все представления признаков для решения задач локализации и классификации.

Чтобы уменьшить эти семантические пробелы, *сеть пирамиды признаков* (feature pyramid network, FPN) добавляет нисходящее соединение от высокоуровневых (семантически более богатых) карт признаков к низкоуровневым (самым бедным семантически) картам признаков, как показано на рис. 2.11d. Это гарантирует, что пирамида признаков имеет сильную семантику на всех уровнях пирамиды. Как и на рис. 2.11с, FPN представляет собой внутрисетевую пирамиду и, следовательно, эффективна, но нисходящие соединения увеличивают ее репрезентативность.

2.8.2.1. Архитектура

Основное различие между FPN и стандартными сетями восходящей классификации, такими как ResNet (He et al., 2016), заключается в добавлении нисходящего пути, позволяющего создавать семантически богатые пирамиды признаков. Этот прием реализуется с помощью простого набора элементов, показанных на рис. 2.14.

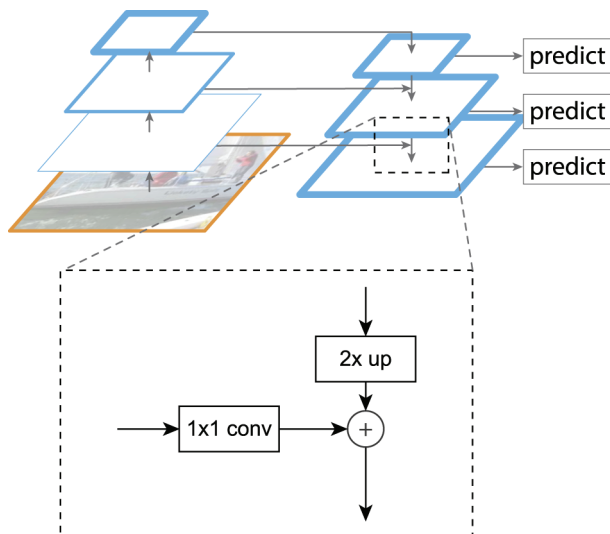


Рис. 2.14 ❖ Строение сети FPN

Восходящий путь

Стандартная сеть с прямой связью, естественно, представляет собой восходящую пирамиду из-за использования операций понижающей дискретизации, таких как пулинг, свертка со страйдом 2 и т. д. В целом разрешение карт признаков уменьшается в 2 раза на каждом этапе сети, где этап определяется как последовательность слоев с одинаковым разрешением. FPN строится на основе ResNet, извлекая восходящую пирамиду из активаций последнего уровня каждой стадии ResNet. В частности, выходные данные слоев conv2, conv3, conv4 и conv5 сети ResNet, обозначенные как $\{C_2, C_3, C_4, C_5\}$, используются для создания пирамиды со страйдом $\{4, 8, 16, 32\}$ пикселя по отношению к входному изображению.

Нисходящий путь и боковые соединения

Цель FPN – обогатить семантику низкоуровневых карт объектов. Простой способ сделать это – добавить высокоуровневые карты объектов строгой семантики к низкоуровневым. Поскольку карты объектов более высокого уровня имеют более низкое разрешение, они сначала увеличиваются в два раза с использованием *выборки ближайшего соседа* (nearest neighbor sampling). Перед суммированием карты объектов нижнего слоя подаются на латеральный сверточный слой 1×1 , чтобы гарантировать, что обе карты объектов имеют одинаковые размеры каналов. Затем карты признаков суммируются поэлементно, и для создания окончательной карты признаков применяется свертка 3×3 , чтобы избежать потенциального размытия из-за операции повышения разрешения. В этих дополнительных слоях нет нелинейности. Процедура повторяется сверху вниз пирамиды, т. е. слоев $\{C_2, C_3, C_4, C_5\}$, чтобы получить окончательную пирамиду FPN из слоев $\{P_2, P_3, P_4, P_5\}$, каждый из которых содержит 256 измерений канала. Каждому слою P_i соответствует слой C_i того же разрешения. Чтобы гарантировать, что все уровни пирамиды FPN обладают одинаковой семантикой, для получения окончательных прогнозов для всех масштабов на разных уровнях используются слои классификации и регрессии ограничивающей рамки.

Поскольку двухэтапные детекторы объектов, такие как рассмотренные выше, нуждаются в пулинге RoI для извлечения признаков экземпляра, обработанных вторым этапом, они не являются полностью сверточными, что усложняет их аппаратную реализацию. Хотя эти детекторы точны, они достигают лишь скорости 10–20 кадров в секунду (fps) на современных графических процессорах. Более высокие скорости обнаружения обычно требуют более дружелюбных к оборудованию архитектур, как правило, полностью сверточных и содержащих один этап. В литературе был предложен ряд таких архитектур, включая YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) и RetinaNet (Lin et al., 2017b). Одноступенчатые детекторы обычно жертвуют точностью ради скорости.

2.9. АРХИТЕКТУРА YOLO

Архитектура YOLO (You only look once) (Redmon et al., 2016) была одной из первых и до сих пор остается самым популярным одноэтапным детектором объектов. Она приобрела популярность в основном благодаря высокой скорости, более 50 кадров в секунду на современном графическом процессоре. Однако ее точность значительно ниже, чем у современных двухкаскадных детекторов. В первой версии YOLO не использовались привязки, как в Faster R-CNN; они появились только в более поздних версиях. Принцип работы YOLO изображен на рис. 2.15. Входное изображение разбивается на ячейки $S \times S$, и для каждой ячейки x делается B прогнозов. Ячейка считается ответственной за объект тогда и только тогда, когда внутри нее находится центр эталонной ограничивающей рамки объекта. Каждый прогноз состоит из четырех координат ограничивающей рамки x , y , w и h и показателя объектности $p(o = 1|x)$. Последнее условие отражает уверенность в том, что предсказанный блок включает в себя объект и идеально равен IoU между предсказанным и эталонным блоками объекта. Если в ячейке не существует объекта, показатель объектности должен быть равен 0. Тогда прогноз достоверности для класса k определяется как

$$p(y = k|x) = p(y = k|o = 1, x)p(o = 1|x), \quad (2.13)$$

где $p(y = k|o = 1, x)$ – классовая условная вероятность того, что класс k появится в ячейке x , при условии что ячейка содержит объект. Однако один набор условных вероятностей класса S является общим для предсказаний B ячейки, т. е. все предсказания в ячейке имеют одинаковые условные вероятности класса. Типичная реализация YOLO для обнаружения 20 классов объектов

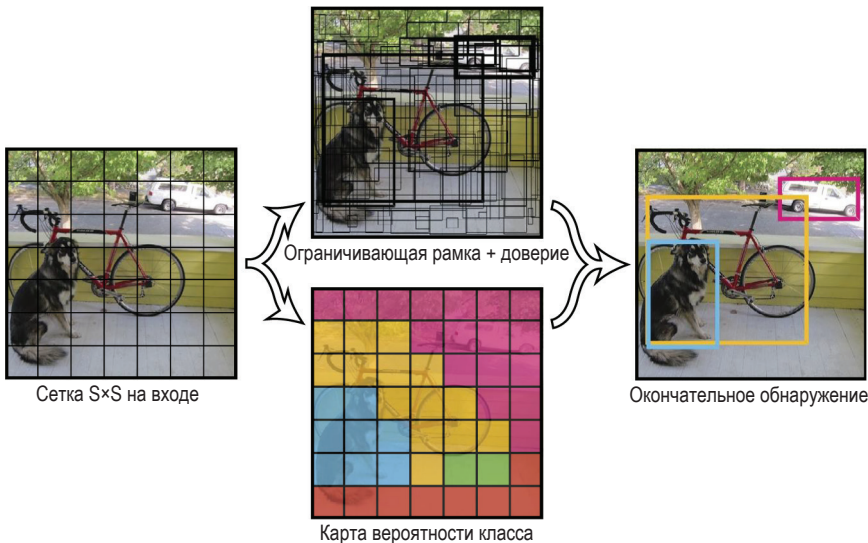


Рис. 2.15 ❖ Принцип работы YOLO

набора данных Pascal VOC (Everingham et al., 2010) использует $S = 7$, $B = 2$ и $C = 20$, всего $7 \times 7 \times 30$ предсказаний.

Базовая структура

Одной из причин эффективности YOLO является базовая сеть, называемая DarkNet. Она опирается на идеи архитектуры GoogLeNet (Szegedy et al., 2015), содержащей 24 сверточных слоя, за которыми следуют 2 полносвязных слоя, выполненных с комбинацией канала 1×1 и сверточных слоев 3×3 , реализация которых оптимизирована для современных графических процессоров.

2.10. Сеть SSD

Сеть SSD (Liu et al., 2016) – это еще один популярный одноступенчатый детектор объектов. Он такой же быстрый, как YOLO, но имеет гораздо более высокую точность, особенно для небольших объектов. Основные отличия заключаются в использовании многомасштабной пирамиды признаков как на рис. 2.11с и привязок RPN.

2.10.1. Архитектура

Базовая сеть представляет собой стандартную сеть классификации изображений (без конечного слоя классификации), например VGG-Net (Simonyan, Zisserman, 2014). К этой сети добавляются некоторые вспомогательные уровни для прогнозирования обнаружения. Общая архитектура показана на рис. 2.16.

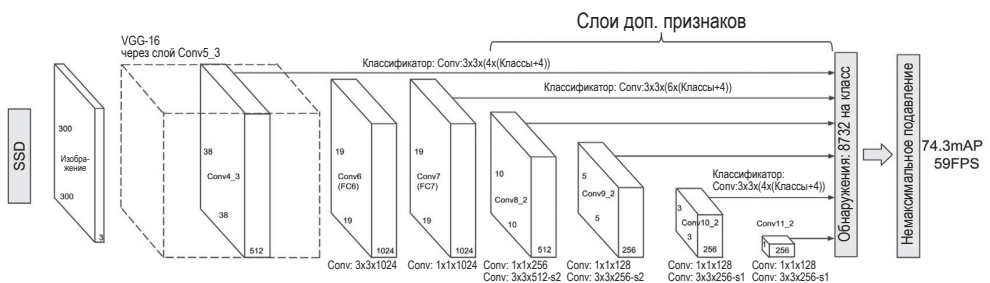


Рис. 2.16 ❖ Устройство сети SSD

Многомасштабное обнаружение

Подобно MS-CNN, SSD использует иерархические представления признаков как на рис. 2.11с для многомасштабного обнаружения. Как показано на рис. 2.16, обнаружения генерируются из карт признаков слоев *Conv4_3*, *Conv6*, *Conv7*, *Conv8_2*, *Conv9_2*, *Conv10_2* и *Conv11_2*, которые имеют разное разрешение. По причинам, рассмотренным выше, это позволяет обнаружи-

вать больше объектов и более точно обнаруживать мелкие объекты. Подобно RPN (Ren et al., 2017), предсказатель SSD, применяемый к каждой сверточной карте признаков, состоит из дополнительного слоя свертки 3×3 , выходными данными которого являются оценки классов и смещения ограничивающей рамки относительно позиций привязки.

Привязки

Подобно RPN, в заданном месте есть k якорей, и для каждой привязки прогнозируются оценки класса c и 4 смещения координат. Следовательно, карта признаков с разрешением $h \times w$ дает $(c + 4) \times k \times h \times w$ выходных данных. Это эквивалентно применению RPN на нескольких картах признаков и помогает понять, почему одноэтапные детекторы в целом менее точны, чем двухэтапные детекторы: они аналогичны стадии генерации предложений последних. При сравнении SSD с RPN в виде реализации MS-CNN основное различие состоит в том, что RPN не зависит от класса, т. е. $c = 2$, в то время как SSD зависит от класса, делая $c + 1$ прогнозов класса, например для набора данных VOC $c = 21$ (Everingham et al., 2010).

2.10.2. Обучение

SSD использует многозадачную функцию потерь, аналогичную (2.5), сочетающую классификацию и регрессию ограничивающей рамки. Для более точного обнаружения она также использует во время обучения жесткий негативный майнинг и сильное расширение данных.

Обработка трудных отрицательных образцов

Сложность обнаружения объектов заключается в том, что большинство отрицательных образцов, например неба, легко классифицировать. Если в обучение включено слишком много простых отрицательных образцов, детектор будет хуже работать с *трудными отрицательными образцами* (hard negatives, отрицательные образцы, которые визуальны похожи на положительные). Эта проблема более серьезна для одноступенчатых детекторов, в которых отсутствует эффективная передискретизация, реализуемая вторым каскадом. *Обработка трудных отрицательных образцов* (hard negative mining) – это механизм выборки, широко применявшийся для обнаружения объектов до появления глубокого обучения (Viola, Jones, 2004; Felzenszwalb et al., 2010; Dollár et al., 2014), предназначенный для решения этой проблемы. Чтобы создать пул отрицательных результатов, SSD сортирует отрицательные образцы по более высоким и более низким показателям достоверности (более высокий означает, что пример сложнее классифицировать) и выбирает лучшие образцы, необходимые для достижения соотношения 1:3 между положительными и отрицательными результатами, аналогично выборке Fast R-CNN. Это заставляет детектор научиться более точно отличать трудные отрицательные образцы.

Дополнение данных

Нехватка обучающих данных – еще одна проблема для обнаружения объектов на основе глубокого обучения, которую можно решить путем *дополнения* (приращения) *данных*. При обучении SSD каждое изображение случайным образом дополняется либо 1) сохранением исходного изображения, либо 2) обрезкой фрагмента минимальной IoU с объектом в $\{0,1, 0,3, 0,5, 0,7, 0,9\}$, либо 3) обрезкой случайного фрагмента. Размер (соотношение сторон) случайного фрагмента выбирается случайным образом в диапазоне $[0,1, 1]$ от исходного размера изображения $\{1/2, 2\}$. После кадрирования размер фрагмента изменяется до фиксированного квадратного размера (например, 512×512) со случайным отражением по горизонтали и отправляется в сеть. Помимо пространственного дополнения, иногда применяются дополнения в цветовом пространстве.

2.11. RETINANET

В то время как обучение на трудных отрицательных примерах устраняет дисбаланс между простыми (например, небо) и трудными объектами, этот подход лишь умеренно эффективен для детекторов глубокого обучения. RetinaNet (Lin et al., 2017b) вместо этого предлагает обобщение кросс-энтропийной потери, обозначаемое как *фокальная*, или *очаговая*, *потеря* (focal loss, FL), которое подавляет легкие отрицательные случаи и подчеркивает трудные.

2.11.1. Фокальная потеря

Функция потерь *перекрестной энтропии* (cross entropy, CE) для бинарной классификации равна

$$CE(p, y) = \begin{cases} -\log(p), & \text{если } y = 1 \\ -\log(1 - p) & \text{в ином случае} \end{cases}, \quad (2.14)$$

где $y \in \{\pm 1\}$ – метка истинности, а $p = p(y = 1|x) \in [0, 1]$ – вероятность для класса $y = 1$. Определим p_t как

$$p_t = \begin{cases} p, & \text{если } y = 1 \\ 1 - p & \text{в ином случае} \end{cases}. \quad (2.15)$$

Следовательно, можно записать $CE(p, y) = CE(p_t) = -\log(p_t)$, как показано на рис. 2.17 (верхняя синяя кривая, где $\gamma = 0$). Можно заметить, что легко классифицируемые примеры (например, с $p_t \gg 0,5$) по-прежнему имеют не-тривиальную величину потерь. Когда большинство обучающих примеров «простые», они доминируют в общих потерях, подавляя вклад «трудных».

Фокальные потери (FL) обобщают потери CE путем добавления коэффициента модуляции $(1 - p_t)^\gamma$ настраиваемого параметра $\gamma \geq 0$:

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t). \quad (2.16)$$

Этот фактор изменяет потери, как показано на рис. 2.17. Если для малых p_t (трудные примеры) потери сильно не меняются, то при больших p_t (простые примеры) они уменьшаются до нуля гораздо быстрее, особенно при больших значениях γ . Следовательно, простые примеры вносят меньший вклад в общие потери. Например, при $\gamma = 2$ простой пример с $p_t = 0,9$ будет иметь в 100 раз меньший вклад при фокальной потере по сравнению с кросс-энтропийной потерей. Величина понижения веса определяется гиперпараметром γ , причем чем больше значение γ , тем сильнее понижение веса, как показано на рис. 2.17. На практике было установлено, что значение $\gamma = 2$ работает лучше всего.

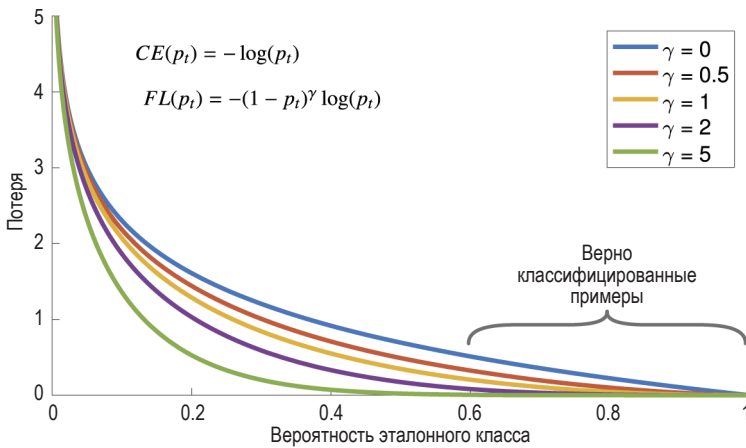


Рис. 2.17 ❖ Визуализация фокальных потерь для различных значений γ . Значение $\gamma > 0$ уменьшает потери легко классифицируемых примеров ($p_t > 0,5$), уделяя больше внимания трудным неправильно классифицированным примерам

В целом фокальные потери уравниваются другим гиперпараметром α :

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t). \quad (2.17)$$

Когда используется эта потеря, проблему подавления простых отрицательных примеров можно обойти во время обучения.

2.12. ПРОИЗВОДИТЕЛЬНОСТЬ ДЕТЕКТОРОВ ОБЪЕКТОВ

Мы закончим эту главу кратким сравнением производительности некоторых из рассмотренных выше детекторов объектов. R-CNN, SPP-Net и Fast R-CNN не включены в это сравнение, поскольку они устарели и редко используются

на практике. В табл. 2.1 представлены сводные данные о производительности остальных методов на наборе данных COCO (Lin et al., 2014) с точки зрения скорости, числа эпох обучения и средней точности (average precision, AP). В COCO показатель AP усредняется по 10 пороговым значениям IoU 0,50:0,05:0,95, а AP_{50} (при пороге 0,5), AP_{75} (при пороге 0,75) AP_S (для малых объектов), AP_M (для средних объектов) и AP_L (для крупных объектов) предоставляют более подробную информацию о производительности. Обратите внимание, что более полная метрика AP, усредненная по порогам IoU 0,50:0,05:0,95, вознаграждает детекторы с лучшей локализацией, чем традиционная метрика AP_{50} при пороге IoU = 0,5. Очевидно, что одноступенчатые детекторы (YOLO и SSD) быстрее, но гораздо менее точны, чем их двухступенчатые аналоги (Faster R-CNN, FPN и Cascade R-CNN). Среди двухкаскадных детекторов скорости сопоставимы, но Cascade R-CNN достигает наивысшей точности. Аналогичные сравнения можно найти в публикациях по обнаружению объектов, и они позволяют практикующим специалистам выбирать детектор с компромиссом между сложностью и точностью, наиболее подходящим для определенного применения.

Таблица 2.1. Характеристики усовершенствованных детекторов объектов на наборе COCO

	Основа	Скорость (fps)	Эпохи	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3	DarkNet-53	48,1	273	33,4	56,3	35,2	19,5	36,4	43,6
SSD512	VGG16	30,7	24	29,4	49,3	31,0	11,7	34,1	44,9
RetinaNet	ResNet-50	24,4	36	38,7	58,0	41,5	23,3	42,3	50,3
Faster R-CNN	ResNet-50	9,6	36	38,4	58,7	41,3	20,7	42,7	53,1
FPN	ResNet-50	26,3	36	40,2	61,0	43,8	24,2	43,5	52,0
Cascade R-CNN	ResNet-50	21,8	36	43,6	61,6	47,4	26,2	47,1	56,9

2.13. ЗАКЛЮЧЕНИЕ

В этой главе мы рассмотрели последние достижения в области обнаружения объектов на основе глубокого обучения. В целом существующие методы можно разделить на две категории: одноэтапные и двухэтапные. Одноэтапные методы быстрее, но менее точны. Два упомянутых поэтапных метода объединяют первую сеть предложений, которая похожа на одноэтапный детектор, но нечувствительна к классам, и вторую сеть, которая классифицирует объекты и уточняет их ограничивающие рамки. За прошедшие годы был сделан большой вклад в улучшение производительности новаторской R-CNN (Girshick et al., 2014) с точки зрения как скорости, так и точности. Сюда относятся концепции, которые имеют важное значение для исследований в области обнаружения объектов, такие как пулинг видимых областей, многозадачные потери, RPN, привязки, каскадное обнаружение и регрессия, многомасштабные представления признаков, методы добавления данных, функции потерь и т. д. Хотя, как показано в табл. 2.1, эти идеи и методы

позволили существенно улучшить производительность, детекторы объектов все еще далеки от совершенства. Также существует большое количество литературы по темам, не затронутым в этом обзоре, таким как сегментация экземпляров, адаптация предметной области или архитектуры глубокого обучения низкой сложности и т. д. Наконец, обнаружение объектов часто используется в качестве предварительного этапа многих других задач компьютерного зрения, включая оценку позы, подписи к изображениям и видео или визуальные ответы на вопросы. Таким образом, хотя современные детекторы объектов на порядки эффективнее тех, что были всего десять лет назад, предстоит еще много исследований по этой проблеме, имеющей фундаментальное значение для компьютерного зрения.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Cai Z., Fan Q., Feris R. S., Vasconcelos N., 2016. A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV, pp. 354–370.
- Cai Z., Vasconcelos N., 2021. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (5), 1483–1498.
- Dalal N., Triggs B., 2005. Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893.
- Dollár P., Appel R., Belongie S. J., Perona P., 2014. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (8), 1532–1545.
- Everingham M., Gool L. J. V., Williams C. K. I., Winn J. M., Zisserman A., 2010. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88 (2), 303–338.
- Felzenszwalb P. F., Girshick R. B., McAllester D. A., Ramanan D., 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (9), 1627–1645.
- Girshick R. B., 2015. Fast R-CNN. In: ICCV, pp. 1440–1448.
- Girshick R. B., Donahue J., Darrell T., Malik J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR, pp. 580–587.
- He K., Zhang X., Ren S., Sun J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV, pp. 346–361.
- He K., Zhang X., Ren S., Sun J., 2016. Deep residual learning for image recognition. In: CVPR, pp. 770–778.
- Krizhevsky A., Sutskever I., Hinton G. E., 2012. Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114.
- Lazebnik S., Schmid C., Ponce J., 2006. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR. IEEE Computer Society, pp. 2169–2178.
- Lin T., Dollár P., Girshick R. B., He K., Hariharan B., Belongie S. J., 2017a. Feature pyramid networks for object detection. In: CVPR. IEEE Computer Society, pp. 936–944.

- Lin T., Goyal P., Girshick R. B., He K., Dollár P.*, 2017b. Focal loss for dense object detection. In: ICCV. IEEE Computer Society, pp. 2999–3007.
- Lin T., Maire M., Belongie S. J., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C. L.*, 2014. Microsoft COCO: common objects in context. In: ECCV, pp. 740–755.
- Liu W., Anguelov D., Erhan D., Szegedy C., Reed S. E., Fu C., Berg A. C.*, 2016. SSD: Single Shot Multibox Detector. ECCV, vol. 9905. Springer, pp. 21–37.
- Lowe D. G.*, 1999. Object recognition from local scale-invariant features. In: ICCV. IEEE Computer Society, pp. 1150–1157.
- Lowe D. G.*, 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.
- Papageorgiou C., Oren M., Poggio T. A.*, 1998. A general framework for object detection. In: ICCV. IEEE Computer Society, pp. 555–562.
- Redmon J., Divvala S. K., Girshick R. B., Farhadi A.*, 2016. You only look once: unified, real-time object detection. In: CVPR, pp. 779–788.
- Ren S., He K., Girshick R. B., Sun J.*, 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6), 1137–1149.
- Rowley H. A., Baluja S., Kanade T.*, 1996. Neural network-based face detection. In: CVPR. IEEE Computer Society, pp. 203–208.
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M. S., Berg A. C., Li F.*, 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115 (3), 211–252.
- Simonyan K., Zisserman A.*, 2014. Very deep convolutional networks for large-scale image recognition. CoRR. arXiv: 1409.1556 [abs].
- Sung K. K., Poggio T. A.*, 1998. Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1), 39–51.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.*, 2015. Going deeper with convolutions. In: CVPR, pp. 1–9.
- Van de Sande K. E. A., Uijlings J. R. R., Gevers T., Smeulders A. W. M.*, 2011. Segmentation as selective search for object recognition. In: ICCV, pp. 1879–1886.
- Viola P. A., Jones M. J.*, 2001. Rapid object detection using a boosted cascade of simple features. In: CVPR. IEEE Computer Society, pp. 511–518.
- Viola P. A., Jones M. J.*, 2004. Robust real-time face detection. International Journal of Computer Vision 57 (2), 137–154.

ОБ АВТОРАХ ГЛАВЫ

Чжаовой Цай (Zhaowei Cai) – ученый и прикладной специалист в Amazon Web Services. Он получил степень бакалавра в области автоматизации в Даляньском морском университете в 2011 г., а также степень магистра и доктора философии в Калифорнийском университете в Сан-Диего в 2019 г. С 2011 по 2013 г. работал научным сотрудником в Институте автоматизации Китайской академии наук. Его текущие исследовательские интересы связаны

с компьютерным зрением и машинным обучением, включая обнаружение и распознавание объектов.

Нуно Васконселос (Nuno Vasconcelos) получил степень бакалавра в области электротехники и компьютерных наук в Университете Порто, Португалия, а также степени магистра и доктора философии в Массачусетском технологическом институте. Он является профессором кафедры электроники и вычислительной техники Калифорнийского университета в Сан-Диего, где возглавляет лабораторию статистических визуальных вычислений. Он получил награду NSF CAREER, стипендию Хеллмана, несколько наград за лучшую научную работу и является членом IEEE.

Глава 3

.....

Обучение с ограниченным подкреплением – статические и динамические задачи

Авторы главы:

Суджой Пол¹, Google Research, Бангалор, Индия;
Амит Рой-Чоудхури, Калифорнийский университет,
Риверсайд, электротехника и вычислительная техника,
Риверсайд, Калифорния, США

Краткое содержание главы:

- уменьшение подкрепления при обучении моделей компьютерного зрения важно для масштабируемости и адаптивности;
- рассмотрены различные методы, которые были предложены для обучения с ограниченным подкреплением, и представлены результаты, подтверждающие эффективность этих методов;
- уделено особое внимание активному обучению для распознавания, слабо подкрепляемому обучению для локализации событий, адаптации предметной области для семантической сегментации и обучению с подкреплением для обнаружения подцелей при обучении роботов динамическим задачам.

¹ Эта глава была написана, когда автор работал в Калифорнийском университете в Риверсайте.

3.1. ВВЕДЕНИЕ

Недавние успехи в компьютерном зрении в основном связаны с использованием огромного массива сложно размеченных данных для обучения моделей распознавания. Но в реальной жизни получение таких больших наборов данных потребует чрезвычайно трудоемкой и дорогостоящей ручной разметки, которая часто выходит за рамки бюджета и может содержать ошибки. Тем не менее множество реальных данных, которые генерируются ежедневно, могут быть использованы с минимальными затратами на разметку или вовсе без нее. Такие данные могут быть неразмеченными или содержать информацию о тегах/метаданных, называемую *слабой разметкой* (weak annotation). Наша цель – разработать методы, которые могут обучать модели распознавания на таких данных с ограниченным объемом ручной работы. В этой главе мы рассмотрим два аспекта обучения с *ограниченным подкреплением* (limited supervision): во-первых, сокращение количества размеченных вручную данных, необходимых для обучения моделей распознавания, и, во-вторых, снижение уровня подкрепления с сильного до слабого, который можно получить из интернета, запросить у оракула или задать как основанные на правилах метки, полученные из знаний предметной области.

В отношении первого аспекта обучения с ограниченным подкреплением мы показываем, что контекстная информация, часто присутствующая в естественных данных, может быть использована для уменьшения количества необходимых аннотаций. В отношении второго аспекта – снижения уровня подкрепления – мы используем слабую разметку вместо плотных сильных меток для обучения задачам плотного прогнозирования. Мы обсудим основы обучения с использованием слабой разметки для обнаружения действий в видео и предметной адаптации моделей семантической сегментации изображений. Все эти обсуждаемые задачи носят статический характер. Продолжая двигаться в направлении обучения со слабой разметкой, мы исследуем последовательные задачи принятия решений, где следующий вход зависит от текущего результата. Мы рассматриваем проблему обучения моделей задачам робототехники на примере небольшого набора человеческих действий путем разложения сложной задачи на подзадачи. Подробное объяснение этих методов следует ниже.

3.2. КОНТЕКСТНО-ЗАВИСИМОЕ АКТИВНОЕ ОБУЧЕНИЕ

В последние годы благодаря прорывному развитию технологий ежедневно генерируется огромное количество визуальных и текстовых данных, которые в основном не размечены для использования в машинном обучении. Кроме того, алгоритмы машинного обучения все чаще встречаются в жизни человека. Большая часть этих алгоритмов основана на обучении с подкреплением, которое требует разметки большого количества данных. Более того,

эти модели необходимо обновлять по мере появления новых данных, чтобы динамически адаптироваться к различным семантическим концепциям, которые могут меняться со временем. Ручная разметка этого непрерывного потока данных является не только утомительной задачей для людей, но и служит причиной ошибок в разметке. В таком случае может быть выгодно размечать только информативные точки данных и пропускать точки данных, несущие избыточную информацию. Обоснованность этой идеи подтверждается в работах (Lapedriza et al., 2013), которые показывают, что не все точки данных несут одинаковое количество информации, и выбор наиболее информативных из них может даже привести к лучшей производительности, чем маркировка всех точек данных в немаркированном наборе. Активное обучение, которое изучалось в литературе в течение последних нескольких десятилетий, показало огромный потенциал в плане выбора информативных точек данных и сокращения усилий по ручной разметке.

3.2.1. Активное обучение

Активное обучение (Settles, 2012) было предложено как решение проблемы сокращения ручной разметки без ущерба для эффективности распознавания. Наглядная иллюстрация принципа активного обучения изображена на рис. 3.1. Используя большой немаркированный набор данных, методы активного обучения сначала выбирают небольшое случайное подмножество данных и обращаются к человеку-эксперту, так называемому *оракулу*, чтобы получить их разметку. Эти точки данных используются для обучения модели прогнозирования согласно поставленной задаче. Это начальная модель. Сле-

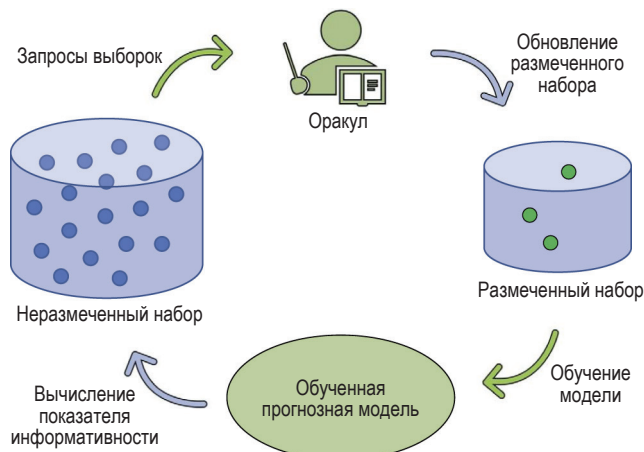


Рис. 3.1 ❖ Схема итеративного процесса в активном обучении. Процесс начинается с извлечения нескольких образцов из немаркированного набора, чтобы запросить у оракула ручную разметку. Затем размеченный набор используется для обновления прогнознй модели, которая далее применяется для вычисления показателя информативности оставшихся неразмеченных данных и, в свою очередь, для выбора небольшого подмножества для следующей разметки

дующая задача состоит в том, чтобы выбрать только небольшое подмножество точек данных из неразмеченного набора, дабы получить максимально возможный объем информации. Показатели полезной информативности этих неразмеченных выборок вычисляются на основе неопределенности текущей модели, плотности данных и т. д. Эти показатели используются при извлечении выборок для разметки. Как правило, активные методы обучения основаны на итерациях с участием человека в цикле, т. е. на вычислении показателей информативности, получении от человека разметки с последующим обновлением модели новыми размеченными точками данных и повторном вычислении показателей информативности немаркированных точек данных, как показано на рис. 3.1. Этот цикл продолжается либо до тех пор, пока бюджет разметки не будет исчерпан, либо постоянно в случае непрерывного обучения, когда концепции могут меняться со временем и требуется постоянное обновление модели.

Обозначения

Прежде чем детально рассмотреть каждый из этих шагов, давайте формализуем обозначения, которые будут использоваться в дальнейшем. Рассмотрим задачу классификации, в которой c категорий. Мы обучаем модель, которая с учетом признака точки данных \mathbf{x} предсказывает функцию распределения вероятности (probability mass function, PMF) $p_\theta(y|\mathbf{x})$ по c категориям, параметризуемую по θ . Здесь θ может быть одним вектором для линейных моделей или группой матриц для глубоких нейронных сетей. Для этого у нас есть размеченное множество кортежей $\mathcal{L} = \{(\mathbf{x}_i, y_i)_{i=1}^l\}$ и неразмеченное множество $\mathcal{U} = \{(\mathbf{x}_j)_{j=1}^u\}$.

Показатели информативности

Большинство методов активного обучения формулируют некий показатель полезности для каждого неразмеченного образца, на основании которого образцы выбираются для ручной маркировки. Одним из наиболее распространенных показателей информативности, упоминаемых в литературе, является *энтропия прогнозов* (Settles, 2012; Li, Guo, 2014; Paul et al., 2016). Для заданной точки данных \mathbf{x} энтропию можно представить следующим образом:

$$H(\mathbf{x}) = \sum_{i=1}^c -p_\theta(y = i|\mathbf{x}) \log p_\theta(y = i|\mathbf{x}). \quad (3.1)$$

Более высокое значение энтропии означает, что классификатор не уверен в прогнозе и, следовательно, должна быть проведена ручная разметка.

Ожидаемое изменение градиентов параметров модели (Settles, 2012) – это еще один показатель полезности, который измеряет величину изменения градиентов, возможных в модели, когда выборка \mathbf{x} включена в обучающую выборку. Он рассчитывается как

$$G(\mathbf{x}) = \sum_{i=1}^c p_\theta(y = i|\mathbf{x}) \|\nabla_\theta l(\mathcal{L} \cup (\mathbf{x}, i))\|_2, \quad (3.2)$$

где $l(., .)$ – функция потерь, используемая для изучения модели классификации. Более высокое значение ожидаемого изменения градиента будет означать большее количество информации, содержащейся в точке данных. Аналогичными показателями, используемыми в литературе, также являются ожидаемое изменение выходных данных модели (Käding et al., 2016) и ожидаемая частота ошибок (Cuong et al., 2013; Li, Guo, 2013).

Плотность данных (Li, Guo, 2013), которая учитывает плотность точек данных в пространстве признаков, является еще одним важным показателем активного обучения. Его можно определить следующим образом:

$$D(\mathbf{x}) = \frac{1}{|nei(\mathbf{x})|} \sum_{\mathbf{x}_i \in nei(\mathbf{x})} 1 - dist(\mathbf{x}, \mathbf{x}_i), \quad (3.3)$$

где $nei(\mathbf{x})$ – соседние точки данных \mathbf{x} . Более высокое значение $D(\mathbf{x})$ будет означать, что точки данных вокруг \mathbf{x} очень близки друг к другу, и, таким образом, получение метки \mathbf{x} будет полезно для понимания окружающих точек данных. Обратите внимание, что этот показатель обычно используется в сочетании с другими показателями, рассмотренными выше. Учитывая эти меры, методы активного обучения выбирают точки данных для ручной аннотации, как обсуждается далее.

Отбор информативных образцов

Выбор информативных образцов в активном обучении зависит от назначения модели и доступного бюджета на разметку. Существует два возможных способа выбора информативных образцов: последовательный, когда за один раз выбирается одна точка данных, или пакетный режим, когда за один раз выбираются несколько точек данных. После выбора образцов модель обновляется, и для неразмеченных точек данных получаются новые оценки. Таким образом, в сценариях, где вычисления ограничены, пакетный режим может оказаться лучше, но последовательный метод может привести к более высокой производительности.

Большинство активных методов обучения считают, что неразмеченный набор данных является фиксированным, что в общем случае может быть не так. Данные могут быть потоковыми, т. е. поступать порциями. В этом случае мы должны выбрать определенную заранее часть для ручной разметки, которая обычно определяется на основе бюджета разметки на пакет. Более сложный, но полезный сценарий – это когда известен общий бюджет разметки и алгоритму необходимо самостоятельно выбрать подходящее подмножество для разметки для каждого пакета потоковых данных. Интересно отметить, что большинство рассмотренных выше методов определяют показатели информативности независимо для выборок без учета взаимосвязей, которые могут возникнуть между точками данных. Кроме того, эти методы учитывают активное изучение одной задачи за раз и не могут выполнять выборку для нескольких задач распознавания одновременно. Они рассматриваются ниже.

3.2.2. Важность контекста активного обучения

В этом разделе мы обсудим, как взаимосвязи между точками данных могут быть полезны для дальнейшего уменьшения подкрепления, необходимого для обучения задачам распознавания. Мы также обсудим, как можно разработать активное обучение для нескольких задач распознавания одновременно.

Контекст в обучении

В реальном мире между точками данных часто возникают отношения. Такие отношения в публикациях нередко называют *контекстом*. Например, рассмотрим отношения между сценами и объектами. Маловероятно найти «корову» в «спальне», но вероятность встретить в одной сцене «кровать» и «светильник» может быть высокой. Таким образом, получение информации о сцене может улучшить предсказание объектов, и наоборот. Точно так же события/действия в видео могут иметь пространственно-временную корреляцию, как показано на рис. 3.2. Обратите внимание на кадры: не зная действий a_2 и a_3 , трудно предсказать, является ли действие a_1 выходом человека из машины или посадкой в машину. Отношения с другими видами деятельности помогают понять эту конкретную деятельность. Отношения также возникают между документами через гиперссылки или цитаты. Несколько работ показали, что во многих приложениях, таких как распознавание действий (Yao, Fei-Fei, 2010; Wang et al., 2013), распознавание объектов (Galleguillos et al., 2008; Choi et al., 2010), классификация текста (Sen, Getoor, 2003; Settles, Craven, 2008) и т. д., отношения между точками данных можно использовать для повышения эффективности распознавания. Во многих из этих работ для целостного понимания используются вероятностные графические модели (Koller, Friedman, 2009), машины структурных опорных векторов (structural support vector machines, SVM) (Cristianini, Ricci, 2008) и т. д. Даже в эпоху глубокого обучения условные случайные поля (Koller, Friedman, 2009) используются для лучшего понимания сцены.

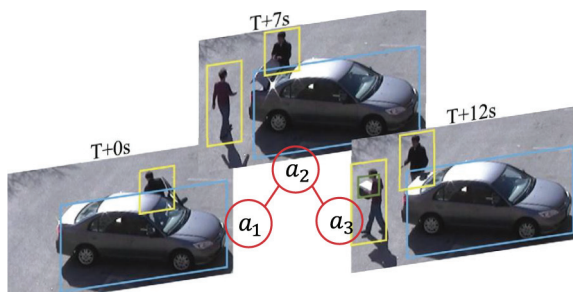


Рис. 3.2 ❖ Последовательность видеопотока из (Oh et al., 2011) представляет три новых неразмеченных действия: человек, выходящий из машины (a_1) в момент $T + 0$ с, человек, открывающий багажник автомобиля (a_2) в момент $T + 7$ с, и человек, несущий предмет (a_3) в момент $T + 12$ с. Корреляция этих действий в пространстве и времени может предоставить контекстуальную информацию для целостного понимания

Контекст для выбора запроса – общая идея

Использование контекста в данных дает нам глобальное понимание сцены. Поэтому интересным направлением исследований является рассмотрение того, может ли контекст также быть полезен для уменьшения количества размеченных выборок. Идея заключается в том, что при наличии контекста, поскольку прогнозы коррелированы, мы можем собрать информацию о большем количестве неразмеченных точек данных, пометая лишь некоторые из них. Как обсуждалось ранее, в отличие от методов, которые не учитывают контекстную информацию в показателях информативности, контекстно-зависимое активное обучение учитывает контекст, чтобы уменьшить количество аннотаций.

Несмотря на то что в некоторых работах рассматриваются отношения между точками данных при активном обучении (Bilgic, Getoor, 2009; Mac Aodha et al., 2014; Hasan, Roy-Chowdhury, 2015; Hu et al., 2013), они не рассматривают поток убеждений между точками данных, чтобы иметь общее понимание их прогнозов, что может быть полезно для выбора наиболее информативных точек. Более того, большинство из них представляют собой алгоритмы, специфичные для конкретной задачи, и имеют дело с активным обучением одной задачи распознавания. Необходим общий подход к активному обучению, учитывающий взаимосвязь между точками данных и который можно использовать в различных предметных областях. Может потребоваться совместное изучение таких задач, как классификация «сцена–объект» (Yao et al., 2012; Wang et al., 2016) или «деятельность–объект» (Jain et al., 2015; Koppula et al., 2013) в активном обучении, чтобы уменьшить усилия по ручной разметке. В таких сценариях сложно выбрать информативные образцы для ручной разметки, поскольку они могут относиться к разным задачам распознавания. Далее мы представляем основу для такого контекстно-зависимого активного обучения, в том числе применимого для нескольких задач.

Контекст для выбора запроса – обзор

Располагая неразмеченным набором, мы описываем схему (Paul et al., 2017), которая выбирает небольшое информативное подмножество точек данных для ручной разметки, используя контекстную информацию, то есть структурные отношения между точками. Логический конвейер данных в рамках данной структуры изображен на рис. 3.3. Мы начинаем с небольшого набора помеченных данных и используем его для построения моделей классификации (\mathcal{C}) и отношений (\mathcal{R}). Здесь \mathcal{R} представляет базовую связь между точками данных через категориальные вероятности совпадения. Обратите внимание, что модели классификации могут содержать несколько классификаторов для нескольких задач распознавания. После обучения первоначальных моделей с учетом новой партии неразмеченных образцов следующий шаг заключается в том, чтобы выбрать подмножество информативных образцов для ручной разметки, которую можно использовать для обновления текущей классификации и моделей отношений.

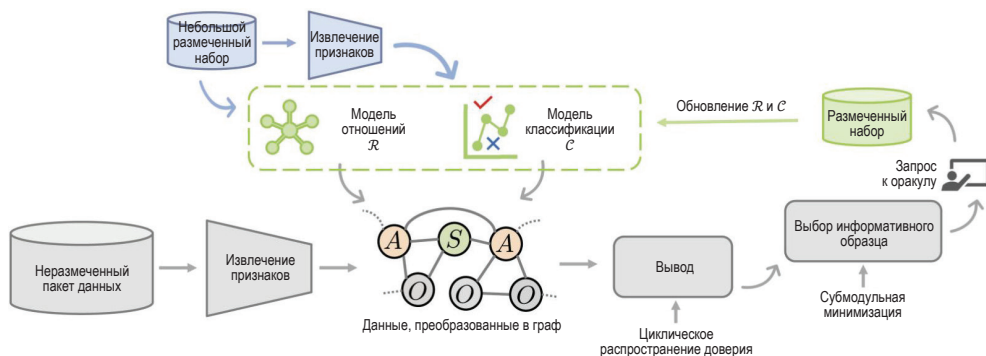


Рис. 3.3 ❖ На этом рисунке представлена схема обсуждаемого метода. 1. Небольшой набор размеченных данных используется для получения исходных отношений (\mathcal{R}) и модели классификации (\mathcal{C}). 2. По мере того как с течением времени становится доступным новый неразмеченный пакет данных, мы сначала извлекаем признаки из необработанных данных. Затем текущие модели \mathcal{R} и \mathcal{C} используются для построения графа на основе данных, чтобы представить отношения между ними. Потом графовый вывод используется для получения вероятностей узлов и ребер, которые, в свою очередь, применяются для формирования информативных выборок, подлежащих ручной разметке. Наконец, вновь размеченные экземпляры используются для обновления моделей \mathcal{R} и \mathcal{C}

По мере появления новых пакетов данных образцы в пакетах разделяются на разные наборы в зависимости от задачи распознавания, к которой они относятся, с последующим выделением признаков. Для каждого неразмеченного образца при помощи текущих классификаторов получают функцию распределения вероятностей по возможным категориям. Она используется вместе с \mathcal{R} для построения графа, узлы которого представляют образцы. Для вычисления меры доверия каждого узла и ребер графов применяется графовый анализ на основе алгоритма обмена сообщениями. Далее выводится теоретико-информационная целевая функция, которая использует меру доверия при выборе наиболее информативных узлов для ручной разметки. Субмодульный характер этой функции оптимизации позволяет нам достичь результата эффективным в вычислительном отношении способом. Новые размеченные узлы используются для обновления моделей \mathcal{R} и \mathcal{C} . Можно отметить, что количество образцов, отбираемых из пакета, переменное и зависит от информативности каждого пакета.

3.2.3. Фреймворк контекстно-зависимого активного обучения

В этом разделе мы представляем фреймворк (набор методов и приемов) для использования контекстной информации, в частности отношений на основе совпадения между точками данных, в среде активного обучения, чтобы уменьшить количество ручных аннотаций для обучения моделей распознавания.

Представление данных

Учтите, что неразмеченные точки данных имеют некоторую базовую структуру, т. е. отношения между ними. Мы используем метод вероятностной графической модели для построения графа, узлы которого представляют неразмеченные образцы, а ребра отражают отношения между ними и работают в качестве путей для потока информации между узлами. Узлы представлены с использованием *потенциалов узлов* (node potential), т. е. прогнозов *функции распределения вероятностей* (probability mass function, PMF) из текущих прогнозных моделей (одиночных или множественных для совместных задач). Ребра представлены с использованием *потенциалов ребер* (edge potential) в виде матрицы совместной встречаемости между категориями, которая фактически является моделью отношений \mathcal{R} . Вычисление совместной встречаемости зависит от конкретного приложения и будет обсуждаться позже. Отметим, что этот фреймворк можно применять к любому приложению, содержащему отношения, которые можно смоделировать как потенциалы ребер.

Построим граф $G = (V, E)$ с экземплярами в \mathcal{U} . Каждый узел в $V = \{v_1, \dots, v_N\}$ представляет каждую точку данных. Ребра $E = \{(i, j) | v_i \text{ и } v_j \text{ связаны}\}$ представляют отношения между точками данных. Потенциалы узлов и ребер назначаются с использованием текущей модели классификации \mathcal{C} и модели отношений \mathcal{R} . Алгоритм передачи сообщений может использоваться для вывода меры доверия узла и ребра, которые представляют собой безусловные вероятности узла и попарное совместное распределение ребер соответственно. Для этой цели можно использовать *алгоритм циклического распространения доверия* (loopy belief propagation, LBP) (Ugm, 2007).

Отбор информативных образцов

На этом этапе цель состоит в том, чтобы, используя вероятности узлов и ребер, выбрать небольшое подмножество $V^* \subset V$ для ручной разметки, что улучшит текущие модели \mathcal{C} и \mathcal{R} . Мы хотим выбрать такое подмножество узлов, чтобы совместная энтропия всех узлов $H(V)$ была минимизирована. Совместная энтропия всех узлов графа может быть аппроксимирована следующим образом:

$$H(V) \approx \sum_{v_i \in V} H(v_i) - \sum_{(i,j) \in E} I(v_j; v_i). \quad (3.4)$$

Заметим, что общая энтропия графа может быть аппроксимирована только для циклического графа в целом, но приведенное выше выражение является точным представлением для ациклических графов. Пусть у нас есть два непересекающихся вершинных подграфа с вершинами V^l и V^{nl} и ребрами, разделенными соответственно на E^l и E^{nl} . Далее, используя уравнение (3.4), мы можем выразить энтропию графа следующим образом:

$$H(V) \approx H(V^l) + H(V^{nl}) - \sum_{\substack{(i,j) \in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i). \quad (3.5)$$

Если мы выберем V^l для ручной разметки, можно показать, что первый и последний члены приведенного выше уравнения становятся равными нулю, что будет уменьшением энтропии. Следовательно, нам нужно выбрать оптимальное подмножество V^{l*} такое, чтобы энтропия была минимизирована как можно сильнее. Задачу оптимизации выбора можно сформулировать следующим образом:

$$V^{l*} = \underset{\substack{V^l \\ \text{так, что } |V^l|=K}}{\operatorname{argmax}} \left[H(V^l) - \sum_{\substack{(i,j) \in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) \right]. \quad (3.6)$$

Вышеупомянутая задача оптимизации является NP-сложной и трудно решаемой. Для эффективного поиска решения можно использовать эвристические методы, такие как *метод ветвей и границ* (branch and bound). Этот метод оптимизации необходим, когда есть строгие бюджетные ограничения на пакет данных, и мы рассмотрели эту проблему в одной из наших работ (Hasan et al., 2018). Однако каждый пакет данных может содержать неоднородный объем информации, и извлечение одинакового количества образцов с ограниченным бюджетом (т. е. K) из каждой партии в целом может быть не очень хорошей идеей. Будет лучше, если удастся определить количество образцов на основе информационного содержания каждой партии. Руководствуясь этой идеей, мы можем изменить приведенную выше задачу оптимизации, чтобы она не ограничивалась регуляризатором количества элементов множества, следующим образом:

$$V^{l*} = \underset{V^l}{\operatorname{argmin}} \left[\sum_{\substack{(i,j) \in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) - H(V^l) + \lambda |V^l| \right],$$

где λ – положительный компромиссный параметр. Можно доказать, что целевая функция в уравнении (3.7) является субмодулярной, что упрощает задачу оптимизации по сравнению с уравнением (3.6). *Минимизация субмодулярных функций* (submodular function minimization, SFM) часто встречается в машинном обучении, теории игр, теории информации и т. д. Подробное описание можно найти в работе Маккормика (McCormick, 2005). Существуют некоторые алгоритмы, которые можно использовать для решения SFM за полиномиальное время. Для решения задачи оптимизации можно использовать популярный алгоритм Minimum Norm Point Фудзихиге–Вольфе (Fujishige et al., 2006).

Обновление модели

После того как выбранные образцы помечены аннотатором-человеком, мы выполняем графовый вывод, обусловленный полученной разметкой, чтобы обновить доверие узлов, а затем применяем концепцию *слабого учителя* (weak teacher) (Чжан и Чаудхури, 2015), которая не вовлекает человека. Мы выбираем узлы, у которых уверенность в классификации больше, чем ϵ , с соответствующей меткой, заслуживающие попадания в размеченное множество \mathcal{L} . Значение ϵ должно быть достаточно высоким, чтобы избежать неправильной разметки. Модель классификации \mathcal{C} обновляется путем переобучения классификатора с использованием \mathcal{L} . Модель \mathcal{R} состоит только из

матрицы совместной встречаемости ψ и увеличивается по мере использования новых размеченных экземпляров.

3.2.4. Практическое применение

В этом разделе мы обсудим несколько задач, при решении которых можно использовать контекстно-зависимое активное обучение, чтобы уменьшить усилия по ручной разметке. В первую очередь разберем три различных применения – совместную классификацию объектов сцены, распознавание действий и классификацию документов. Как показано на рис. 3.4, у приложений есть данные, которые имеют общие отношения между собой. Мы используем линейные классификаторы, такие как машина опорных векторов (SVM) (Chang and Lin, 2011), в качестве базового классификатора во всех приложениях, обсуждаемых далее. При активном обучении результаты применения этого метода обычно сравниваются с качеством классификации на базе полного набора, т. е. с качеством, достигаемым, когда все точки данных (кроме тестового набора) размечены и используются для обучения. Мы также сравниваем наш метод активного обучения с другими популярными методами активного обучения, представленными в научных публикациях, например с пакетным ранжированием (Batch Rank, Chakraborty et al., 2015), BvSB (Li et al., 2012), энтропией (Entropy, Settles, 2012; Holub et al., 2008), выборкой на основе плотности (Density-Based Sampling, DENS) (Settles, 2012), ожидаемой длиной градиента (Expected Gradient Length, GRL) (Settles and Craven, 2008) и случайной выборкой (Random Sampling).

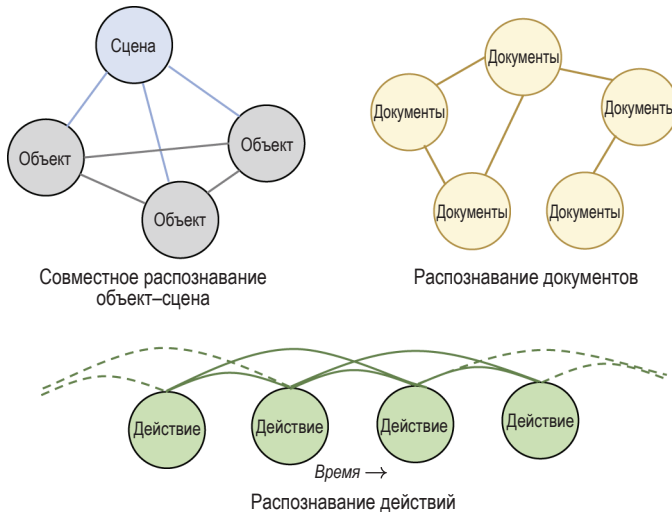


Рис. 3.4 ❖ Семантические графы контекста для трех задач, обсуждаемых в разделе 3.2.4, – совместное распознавание сцены и объекта, когда мы используем контекст, присутствующий между сценой и объектами на изображении, распознавание документа, когда информация о контекстных связях используется совместно через цитаты/веб-ссылки, и распознавание деятельности, когда последовательности действий имеют общие пространственно-временные отношения

Классификация сцены и объектов

Сцена и объекты обычно связаны контекстом и вместе встречаются в изображениях. Хотя классификаторы сцен и объектов представлены по отдельности, их совместное понимание можно использовать в рамках активного обучения для сокращения объема ручной разметки (Yao et al., 2012). Это особенность нашего фреймворка, так как он может активно обучаться с несколькими задачами распознавания одновременно. Набор данных SUN (Choi et al., 2010; Xiao et al., 2010) хорошо подходит для экспериментов с этим фреймворком, поскольку он содержит аннотации как для всей сцены, так и для объектов сцены. Мы извлекаем признаки из предварительно обученных сетей VGG-net (Zhou et al., 2014) и Alex-net (Krizhevsky et al., 2012) для сцен и объектов соответственно. Далее применяем выборочный поиск, используемый в конвейере RCNN, для получения предложений (гипотез) объектов. Как показано на рис. 3.4, при совместном распознавании сцены и объектов мы представляем каждое изображение в виде графа с одним узлом сцены и несколькими узлами объектов, соответствующими различным предложениям объектов на изображении. Граф считается полносвязным с двумя разными типами ребер – сцена–объект и объект–объект. Потенциал ребер для связей сцена–объект вычисляется как частота совместной встречаемости категории сцены с категорией объекта; а для связей объект–объект потенциал ребер представляет собой частоту совместной встречаемости категорий объектов в изображении.

В случае классификации сцен контекстно-зависимый метод активного обучения требует только 35 % ручной разметки для достижения практически 100%-ной точности классификации. Другие методы, описанные в литературе, требуют около 60 % ручной разметки для получения аналогичной точности. При распознавании объектов метод контекстно-зависимого активного обучения требует 45 % ручной разметки, тогда как методы, описанные в литературе, требуют 65 % разметки для достижения аналогичной точности (почти 100 %). Наша работа (Varpu et al., 2016) по адаптации модели распознавания сцены и объектов демонстрирует результаты, специфичные для этого приложения.

Классификация документов

Документы, как правило, связаны между собой цитатами и гиперссылками, которые можно использовать с помощью нашего метода активного обучения, чтобы сократить усилия по ручной разметке. Для наших экспериментов по классификации документов мы используем набор данных CORA (Sen et al., 2008). Он состоит из 2708 научных публикаций, разделенных на семь категорий. В нем представлено 5429 ссылок (цитирований) между публикациями. В свою очередь, публикации представлены с помощью словаря из 1433 уникальных слов, а векторы признаков $\in \{0, 1\}^{1433}$ указывают на отсутствие или

наличие этих слов. Как показано на рис. 3.4, при классификации документов мы представляем все документы как узлы графа, которые связаны, если один документ цитирует другой документ. Мы рассматриваем потенциал ребер как частоту совместной встречаемости, когда публикация категории i цитирует публикацию категории j .

Для классификации документов контекстно-зависимому методу активного обучения требуется всего 33 % ручных аннотаций, чтобы достичь практически полной достоверности (Paul et al., 2017). Другие методы, описанные в литературе, требуют около 50 % ручной разметки для получения аналогичной точности. Это свидетельствует о том, что использование контекстной информации помогает уменьшить объем ручной разметки за счет использования сопутствующей информации, которая легко извлекается из наборов целевых данных.

Классификация действий

Последовательные *действия*, как правило, связаны в пространстве-времени, что можно использовать для уменьшения количества экземпляров, выбранных для ручной разметки. Для наших экспериментов по классификации действий мы используем набор данных VIRAT (Oh et al., 2011) о деятельности человека. Набор данных состоит из 11 видеороликов, разделенных на 329 последовательностей действий. Мы извлекли признаки, используя предварительно обученную модель трехмерных сверточных сетей (Tran et al., 2015). Мы извлекаем признаки для 16 кадров за раз с временным страйдом 8, а затем применяем max-пулинг по временному измерению, чтобы получить один вектор для каждого действия. Мы считаем, что существует связь между двумя действиями, если они происходили в пределах определенного пространственно-временного расстояния. Мы рассматриваем потенциал ребер как совместную пространственно-временную встречаемость двух действий.

В случае классификации деятельности метод активного обучения с учетом контекста требует лишь около 18 % ручной разметки для достижения почти полной точности. Другие методы, описанные в литературе, требуют около 40 % ручной разметки для достижения аналогичной точности. Обратите внимание, что, хотя в этом эксперименте мы используем отношения между *категориями* действий, мы также можем использовать контекстную информацию, передаваемую *объектами*, вовлеченными в действия, что особенно важно в действиях, связанных с взаимодействием человека и объекта. На рис. 3.5 представлен пример, иллюстрирующий работу контекстно-зависимого метода активного обучения. Как видно из схемы, получение знаний о некоторых узлах помогает уменьшить энтропию других узлов, тем самым снижая затраты на ручную разметку.

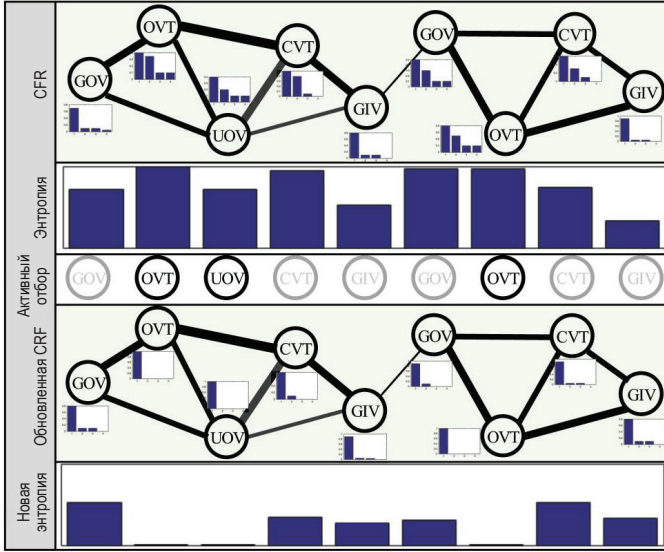


Рис. 3.5 ❖ Пример реализации предлагаемой нами системы активного обучения на части последовательности действий из набора данных VIRAT (Oh et al., 2011). Круги – это узлы действий вместе с их распределением вероятностей классов. Ребра имеют разную толщину в зависимости от попарной взаимной информации. Метки узлов: выход из автомобиля (GOV), открытие багажника (OVT), выгрузка из багажника (UOV), закрытие багажника (CVT) и посадка в автомобиль (GIV). Статистический анализ графа (вверху) дает нам предельное распределение вероятностей узлов и ребер. Мы используем эти распределения для вычисления энтропии и взаимной информации. Относительная взаимная информация показана толщиной ребер, а энтропия узлов нанесена под верхней моделью CRF¹. Уравнение (3.14) использует критерии энтропии и взаимной информации для выбора наиболее информативных узлов (2-OVT, 3-UOV и 7-OVT). Мы обуславливаем эти узлы (они выделены цветом) и снова выполняем вывод, что обеспечивает более точное распознавание и систему с более низкой энтропией (нижний граф)

3.3. Локализация событий при слабой РАЗМЕТКЕ

Временная локализация активности является ключевой проблемой компьютерного зрения, где при наличии длинного видео алгоритм должен локализовать во времени части видео, соответствующие различным интересующим

¹ CRF (conditional random field, условное случайное поле) – это графовая модель для классификации, в которой у вас есть два штрафа: один для классификации узлов и другой для ребер, когда штрафуются несогласованность соседних узлов. Характерным отличием этого метода является возможность учитывать контекст классифицируемого объекта. – Прим. перев.

категориям событий (Aggarwal, Ryoo, 2011). Недавний успех в решении этой задачи (Xu et al., 2017; Zhao et al., 2017) основан на *полном* обучении, которое нуждается в покадровой разметке действий. Однако получение такой точной покадровой информации требует огромного ручного труда. Такой метод плохо масштабируется при увеличении числа камер и категорий действий. С другой стороны, человеку гораздо проще указать несколько категориальных меток, которые инкапсулируют содержание видео. Кроме того, видео, доступные в интернете, часто сопровождаются тегами, обеспечивающими семантическую дискриминацию. Такую разметку на уровне видео обычно называют *слабой* разметкой (weak labels), и ее можно использовать для обучения моделей с возможностью классификации и локализации действий в видео, как показано на рис. 3.6.

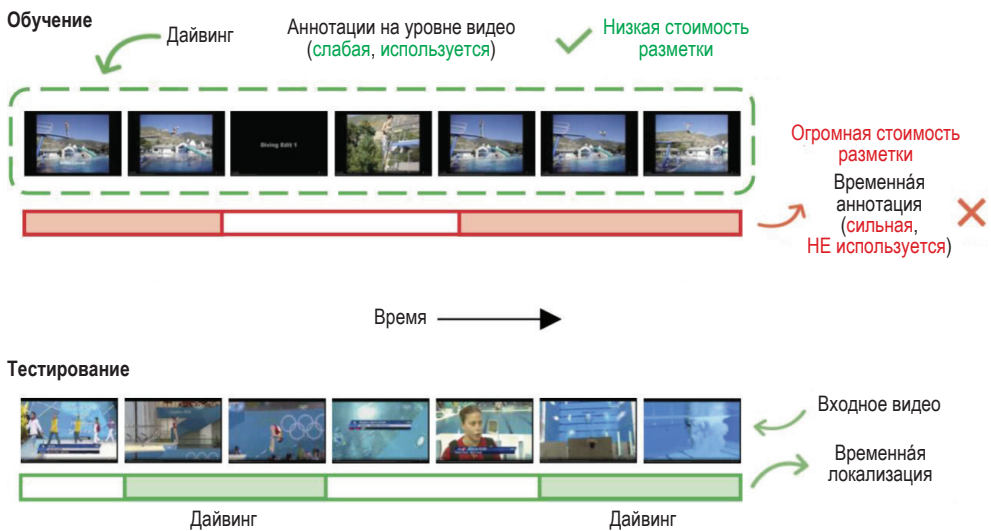


Рис. 3.6 ❖ Здесь представлен протокол последовательных испытаний локализации действия со слабой разметкой. Обучающий набор состоит из видео с тегами активности на уровне видео, а НЕ времени. Стоит отметить, что во время тестирования сеть не только предлагает метки действий в видео, но и определяет их местонахождение во времени

В компьютерном зрении исследователи использовали слабую разметку при обучении моделей для нескольких задач, включая семантическую сегментацию (Hartmann et al., 2012; Khoreva et al., 2017; Yan et al., 2017), визуальное отслеживание (Zhong et al., 2014), реконструкцию (Tulyakov et al., 2017; Kanazawa et al., 2016), обобщение видео (Panda et al., 2017), обучение роботизированным манипуляциям (Singh et al., 2017), субтитрирование видео (Shen et al., 2017), границы объектов (Khoreva et al., 2016), распознавание мест (Arandjelovic et al., 2016) и т. д. Проблема локализации со слабой разметкой аналогична слабому обнаружению объектов на изображениях, где метки категорий объектов предоставляются на уровне изображения. В этой области было проведено несколько исследований, в которых в основ-

ном использовались методы *многовариантного обучения* (multiple instance learning, MIL) (Чжоу, 2004) из-за их тесной связи с точки зрения структуры информации, доступной для обучения. Положительные и отрицательные пакеты, необходимые для MIL, генерируются с помощью современных методов предложения регионов (Li et al., 2016; Jie et al., 2017). Несмотря на сходство, временная локализация с использованием слабой разметки является гораздо более сложной задачей по сравнению с аналогичным обнаружением объектов. Основная причина заключается в дополнительных различиях в содержании, а также в протяженности видео по временной оси. В нескольких работах (Bojanowski et al., 2015; Huang et al., 2016) рассматривалась доступность во время обучения информации о временном порядке действий, помимо разметки на уровне видео.

Далее мы формально опишем задачу временной локализации со слабой разметкой, а затем представим ее решение.

Постановка задачи. Предположим, что у нас есть обучающая выборка из n видео $X = \{x_i\}_{i=1}^n$ с переменной продолжительностью во времени, обозначаемой $L = \{l_i\}_{i=1}^n$ (после извлечения признаков), и набор меток действий $\mathcal{A} = \{a_i\}_{i=1}^n$, где $a_i = \{a_i^j\}_{j=1}^{m_i}$ – метки $m_i (\geq 1)$ для i -го видео. Мы также определяем множество категорий действий как $\mathcal{S} = \bigcup_{i=1}^n a_i = \{\alpha_i\}_{i=1}^C$. Во время тестирования, исходя из видео x , нам нужно предсказать множество $x_{det} = \{(s_j, e_j, c_j, p_j)\}_{j=1}^{n(x)}$, где $n(x)$ – количество обнаружений для x . Элементы s_j, e_j – время начала и время окончания j -го обнаружения, c_j представляет прогнозируемую категорию действия с достоверностью p_j . На рис. 3.7 представлен обзор фреймворка для временной локализации событий со слабой разметкой, детали которого последовательно обсуждаются далее.

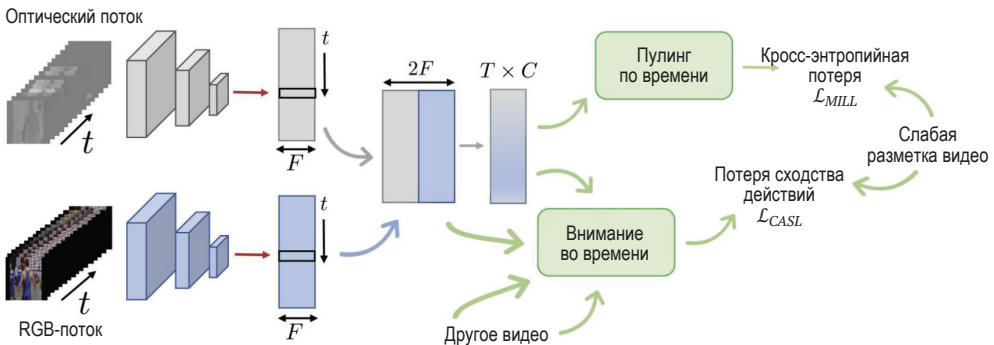


Рис. 3.7 ❖ На этом рисунке представлен предлагаемый нами фреймворк для локализации и классификации действий со слабой разметкой. Для имеющегося видео мы извлекаем признаки из двух потоков – RGB и оптического (optical flow). После объединения векторов признаков из двух потоков мы обучаем несколько слоев, характерных для задачи слабой локализации, и, наконец, проецируем в пространство категорий, чтобы получить матрицу $T \times C$, где T и C – количество временных шагов и категорий соответственно. Мы используем две функции потерь для обучения параметров сети – кросс-энтропийную потерю для прогнозов с пулингом во времени и потерю *сходства совместных действий* (coactivity similarity loss), полученную с использованием пары видеороликов, содержащих по крайней мере одну общую категорию

3.3.1. Архитектура сети

Особое внимание мы уделяем *двухпотокowym сетям* (two-stream network), поскольку они инкапсулируют информацию как о признаках внешнего вида в потоке RGB, так и о признаках движения в оптическом потоке. Для извлечения признаков мы используем две сети – UntrimmedNets (Wang et al., 2017), предварительно обученную на Imagenet, и I3D (Carreira, Zisserman, 2017). Обратите внимание, что остальная часть нашего фреймворка не зависит от используемых признаков. Количество кадров, отправляемых в качестве входных данных в эти сети, зависит от их архитектуры. Поток RGB сети UntrimmedNets принимает 1 кадр в качестве входных данных, тогда как его оптический поток занимает пять кадров для каждого вектора признаков. В случае сети I3D и RGB, и оптический потоки занимают 16 кадров для каждого вектора признаков.

Обычные видеоролики могут иметь большие различия по продолжительности: от нескольких секунд до более часа. В системе со слабой разметкой мы располагаем информацией о метках для видео в целом, что требует обработки всего видео сразу. Это может быть проблематично для очень длинных видео из-за ограничений памяти графического процессора. В качестве решения данной проблемы мы отправляем на вход все видео целиком, если его длина меньше предопределенной длины T , определяемой пропускной способностью графического процессора. Однако если длина видео больше T , мы случайным образом извлекаем из него клип длиной T со смежными кадрами и присваиваем извлеченному видеоклипу все метки всего видео. Можно отметить, что, хотя это может привести к некоторым ошибкам в разметке, данный способ выборки имеет преимущества дополнения данных и хорошо работает на практике.

После извлечения признаков видео из двухпотокowych сетей мы получаем матрицу размерности $\mathbf{X}_i \in \mathbb{R}^{T \times 2F}$, где T – количество временных шагов видео, а F – размерность признаков потоков, которые объединяются для получения признаков с размерностью $2F$. Затем мы пропускаем эти функции через полносвязный слой с нелинейностью ReLU и отсеком, после чего следует слой классификации, который дает нам окончательную матрицу категориальных прогнозов $\mathcal{A} \in \mathbb{R}^{T \times C}$, где C – количество категорий.

3.3.2. k-мат множественное обучение

Задача локализации и классификации действий со слабой разметкой, описанная выше, может быть напрямую сопоставлена с задачей *многоязычного обучения* (multiple instance learning, MIL) (Чжоу, 2004). В MIL отдельные образцы группируются в две *корзины*, а именно положительные и отрицательные. Положительная корзина содержит по крайней мере один положительный экземпляр, а отрицательная корзина точно не содержит положительного экземпляра. Используя эти корзины в качестве обучающих данных, нам нужно обучить модель, которая сможет классифицировать как положительный или отрицательный каждый экземпляр, помимо классификации корзин. В нашем

случае мы рассматриваем все видео как набор экземпляров, где каждый экземпляр представлен вектором признаков в определенный момент времени. Чтобы вычислить потери для каждой корзины, т. е. видео в нашем случае, нам нужно представить каждое видео, используя один показатель достоверности для каждой категории.

Для определенного видео мы вычисляем показатель активации, соответствующий определенной категории, как среднее значение k -тах активации по временному измерению для этой категории. Поскольку количество видеороликов в корзине сильно различается, мы устанавливаем k пропорциональным количеству элементов в корзине. После этого применяем нелинейность softmax для получения функции распределения вероятности по всем категориям, что позволяет вычислить вектор прогнозов p_i по категориям. Нам нужно сравнить эту PMF с эталонным распределением меток для каждого видео, чтобы вычислить потери MIL (MIL loss, MILL). Поскольку каждое видео может содержать несколько действий, мы представляем вектор меток для видео, где во временных позициях стоят единицы, если это действие происходит в видео, иначе стоят нули. Затем мы нормализуем этот эталонный вектор истинности, чтобы преобразовать его в действительный PMF. Таким образом, MILL представляет собой перекрестную энтропию между предсказанной PMF p_i и эталоном, которую затем можно представить следующим образом:

$$\mathcal{L}_{MILL} = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^C -y_i^j \log(p_i^j), \quad (3.8)$$

где $y_i = [y_i^1, \dots, y_i^C]^T$ – нормализованный эталонный вектор, представляющий слабые метки.

3.3.3. Сходство совместных действий

Потеря сходства совместных действий (coactivity similarity loss, CASL) изменяет ограничения для лучшего обучения сетевых параметров в задаче локализации действий. Задача *временной локализации и классификации действий* (weakly supervised temporal activity localization, W-TALC) побуждает нас выявлять корреляции между видео похожих категорий. Прежде чем раскрыть этот вопрос более подробно, определим специфические для категории множества для j -й категории как $\mathcal{S}_j = \{\mathbf{x}_i | \exists a_i^k \in \mathbf{a}_i \text{ так, что } a_i^k = \alpha_j\}$, т. е. множество \mathcal{S}_j содержит все видео обучающего набора, одной из меток которого является действие α_j . В идеале нам могут понадобиться следующие свойства представлений изученных признаков \mathbf{X}_i , которые обсуждались в разделе 3.3.1:

- видеопара, принадлежащая множеству \mathcal{S}_j (для любого $j \in \{1, \dots, C\}$), должна иметь сходные представления признаков в частях видео, где происходит действие α_j ;
- для одной и той же видеопары представление признаков части, где α_j встречается в одном видео, должно отличаться от представления другого видео, где α_j не встречается.

Эти свойства не применяются напрямую в MILL. Поэтому мы вводим CASL, чтобы встроить желаемые свойства в изученные представления признаков. Поскольку у нас нет кадровых меток, мы используем категориальные активации \mathcal{A} для определения необходимых частей действий. Функция потерь разработана таким образом, что помогает одновременно изучать представление признака и проекцию пространства меток. Сначала мы нормализуем категориальные активации видео вдоль временной оси, используя нелинейность softmax, чтобы получить $\hat{\mathcal{A}}$ во время t и категорию j как $\hat{\mathcal{A}}_i[j, t] = \exp(\mathcal{A}_i[t, j]) / \sum_{t'=1}^{l_i} \exp(\mathcal{A}_i[t', j])$, где t обозначает моменты времени и $j \in \{1, \dots, C\}$. Мы называем их *вниманием* (attention), поскольку они относятся к частям видео, где происходит действие определенной категории. Высокое значение внимания для определенной категории указывает на высокую вероятность появления этой категории. Чтобы сформулировать функцию потерь, давайте сначала определим категориальные векторы признаков регионов с высоким и низким вниманием следующим образом:

$$\begin{aligned} {}^H f_i^j &= \mathbf{X}_i \hat{\mathcal{A}}_i[:, j]; \\ {}^L f_i^j &= \frac{1}{l_i - 1} \mathbf{X}_i (\mathbf{1} - \hat{\mathcal{A}}_i[:, j]), \end{aligned} \quad (3.9)$$

где ${}^H f_i^j, {}^L f_i^j \in \mathbb{R}^{2048}$ представляет агрегированные представления признаков области высокого и низкого внимания соответственно в видео i для категории j . Чтобы реализовать два свойства, упомянутых выше, мы используем ранжирование кусочно-линейной функцией потерь (ranking hinge loss). Для видеопары $\mathbf{x}_m, \mathbf{x}_n \in \mathcal{S}_j$ функция потерь может быть представлена следующим образом:

$$\begin{aligned} \mathcal{L}_j^{mn} &= \frac{1}{2} \{ \max(0, d[{}^H f_m^j, {}^H f_n^j] - d[{}^H f_m^j, {}^L f_n^j] + \delta) \\ &\quad + \max(0, d[{}^H f_m^j, {}^H f_n^j] - d[{}^L f_m^j, {}^H f_n^j] + \delta) \}, \end{aligned} \quad (3.10)$$

где $d[]$ – косинусное расстояние, а δ – параметр поля (margin parameter), и в наших экспериментах мы установили его равным 0,5. Два члена в функции потерь эквивалентны по смыслу, и они означают, что функции области высокого внимания в обоих видео должны быть более похожими, чем функция области высокого внимания в одном видео и функция области низкого внимания в другом видео. Сводные потери (total loss) для всего обучающего набора вычисляются для каждой пары видео, имеющей хотя бы одну общую категорию. Для обучения сети две функции потерь в уравнениях (3.8) и (3.10) могут быть оптимизированы совместно.

Локализация. После обучения весов сети мы используем их для локализации событий в видео во время тестирования. Для конкретного видео мы получаем оценки \mathcal{A} достоверности по категориям. Для каждой категории мы получаем порог, который является средней точкой между максимальной и минимальной активациями для этой категории, и используем его как порог активаций при получении локализаций.

3.3.4. Практическая реализация

В этом разделе мы исследуем эффективность предлагаемого фреймворка для локализации действий во времени и классификации видео со слабой маркировкой. Сначала обсудим наборы данных, а затем детали реализации, количественные и некоторые качественные результаты.

Наборы данных

Мы проводим экспериментальный анализ двух наборов данных, а именно ActivityNet v1.2 (Heilbron et al., 2015) и Thumos14 (Idrees et al., 2017). Эти два набора данных содержат необрезанные видеоролики с пок кадровыми метками действий, происходящих в видео. Однако, поскольку наш алгоритм ориентирован на слабую разметку, мы используем только теги действий, связанные с полным видео. Набор данных ActivityNet v1.2 содержит 4819 видео для обучения и 2383 видео для проверки, которые мы используем для тестирования. Количество упоминаемых категорий составляет 100; в среднем приходится 1,5 временных сегмента деятельности на одно видео. Набор данных Thumos14 содержит 200 видеороликов, которые мы используем для обучения, и 212 тестовых видеороликов, разделенных на 20 категорий. Среди этих видеороликов 200 обучающих и 213 тестовых имеют разметку, относящуюся к 20 категориям. Хотя это меньший набор данных, чем ActivityNet1.2, временные метки очень точны и содержат в среднем 15,5 временного сегмента деятельности на видео. В этом наборе данных есть несколько видеороликов, в которых происходит несколько действий, что делает его еще более сложным. Продолжительность видео также широко варьируется от нескольких секунд до более часа. Меньшее количество видео затрудняет эффективное обучение со слабой разметкой.

Локализация действий

Для сравнения качества локализации действий на шкале времени используется *математическое ожидание средней точности* (mean average precision, mAP) при различных пороговых значениях IoU между прогнозируемой и истинной локализациями. Для Thumos14 мы обсуждаем среднее значение mAP для порогов $\text{IoU} \in \{0, 1, 0, 2, 0, 3, 0, 4, 0, 5\}$. Результаты представлены в табл. 3.1, разделенной на три строки: (а) методы с использованием сильной разметки, т. е. с использованием временных аннотаций для каждого действия, появляющегося в видеороликах, (б) методы со слабой разметкой, предложенные другими авторами, и, наконец, (в) результаты метода, предложенного нами (Paul et al., 2018). Выяснилось, что даже со слабой разметкой, которую гораздо проще получить, наш алгоритм демонстрирует результаты, близкие к чрезвычайно трудоемким методам с сильной разметкой. Важно отметить, что хотя предварительно обученные признаки I3D (I3DF) Kinetics имеют некоторые знания о действиях, использование только MILL, как в (Wang et al., 2017), в сочетании с I3DF работает намного хуже, чем в сочетании с CASL, а именно 33,1 против 39,8. В случае ActivityNet v1.2 наш метод дает в среднем 18, тогда как методы, использующие строгую разметку, такие как SSN, дают 24,8.

Таблица 3.1. Сравнение качества обнаружения на Thumos14. UNTF и I3DF – это сокращения для признаков UntrimmedNet (предварительно обученные признаки ImageNet) и признаков I3D соответственно

Обучение	Методы	Среднее IoU 0,1:0,1:0,5
Сильное	R-C3D (Xu et al., 2017)	43,1
	SSN (Zhao et al., 2017b)	47,4
	UntrimmedNets (Wang et al., 2017)	29,0
Слабое	STPN (UNTF) (Nguyen et al., 2018)	30,9
	STPN (I3DF) (Nguyen et al., 2018)	34,9
Слабое (Paul et al., 2018)	MILL+CASL+UNTF	33,8
	MILL+CASL+I3DF	39,8

Классификация действий

Здесь мы представляем качество классификации видов деятельности при помощи предложенного нами фреймворка. Мы также используем математическое ожидание средней точности (mAP) для вычисления качества классификации на основе предсказанных оценок уровня видео p после применения softmax. Результаты представлены в табл. 3.2. Они свидетельствуют о том, что предлагаемый метод (Paul et al., 2018) работает значительно лучше, чем другие современные методы классификации видео. Это может быть частично связано с используемыми признаками, но, что наиболее важно, с тем, как происходит обучение в методах локализации действий со слабой разметкой, которые игнорируют фоновые области при классификации видео с содержащимися в них действиями.

Таблица 3.2. Сравнение качества классификации на Thumos14. UNTF и I3DF – это сокращения для признаков UntrimmedNet (предварительно обученные признаки ImageNet) и признаков I3D соответственно

Обучение	Методы	Thumos14	ActivityNet-1.2
Сильное	TSN (Wang et al., 2016b)	72,0	86,3
	UntrimmedNets (Wang et al., 2017)	82,2	91,3
Слабое	MILL+CASL (Paul et al., 2018)	85,6	93,2

Качественные результаты

Мы представляем несколько интересных примеров локализации с эталонами на рис. 3.8. На этом рисунке показаны два примера из Thumos14 и два из набора данных ActivityNet1.2. Чтобы проверить, как предлагаемый фреймворк работает с видео за пределами вышеупомянутых наборов данных, мы протестировали обученные сети на случайно выбранных видео с YouTube. Представляем два таких примера обнаружения на рис. 3.8, используя модель, обученную на Thumos14.

Первый пример на рис. 3.8 довольно сложен, поскольку локализация должна точно соответствовать частям видео, где происходит замах в гольфе, признаки которого в области RGB очень похожи на части видео, где игрок

готовится к замаху. Несмотря на это, наша модель способна локализовать соответствующие части замаха, возможно, на основе признаков потока. Во втором примере из Thumos14 обнаружение крикетного шота (cricket shot) и крикетного боула (cricket bowl) коррелирует во времени. Это связано с тем, что шот и боул¹ – это два действия, которые в видео про крикет обычно встречаются одновременно. Чтобы научить модель выполнять точную локализацию подобных действий, требуются видеоролики, в которых есть только одно из этих действий. Однако в наборе данных Thumos14 очень мало обучающих примеров содержат только одно из этих двух действий, что объясняет поведение, отмеченное на рисунке.



Рис. 3.8 ❖ На этом рисунке представлены некоторые результаты качественного анализа на наборах Thumos14, ActivityNet1.2 и нескольких случайных видео с YouTube

¹ Shot – это удар, при помощи которого боулер отражает бросок (bowl). Поэтому в большинстве видео боул и шот присутствуют практически одновременно. В профессиональном крикете выделяют 17 разновидностей шота (к вопросу о сложности распознавания действия). – Прим. перев.

В третьем примере, взятом из ActivityNet1.2, хотя игра в поло встречается в первой части видео, эта метка отсутствует в эталонном наборе. Однако наша модель способна локализовать даже такие сегменты деятельности! Тот же вывод применим и к четвертому примеру, где реальная волейболка встречается в кадрах редко, но отклик нашей модели точно совпадает с ее появлением, хотя ошибочные эталонные метки разбросаны почти по всему видео. Эти два примера красноречиво подтверждают необходимость использования локализации со слабой разметкой, поскольку получение точных и единодушных эталонов от нескольких специалистов по маркировке является трудным, дорогостоящим, а иногда даже неосуществимым занятием.

Пятый пример основан на случайно выбранном видео с YouTube. В нем есть человек, который жонглирует мячиками на открытом воздухе. Но большинство видео в Thumos14 той же категории сняты в помещении, а человек занимает значительную часть кадра в пространстве. Несмотря на такие различия в данных, наша модель способна локализовать некоторые части деятельности. Тем не менее модель также предсказывает, что некоторые части видео относятся к жонглированию футбольным мячом, что может быть связано с тем, что обучающие образцы в Thumos14 содержат комбинацию движений ног, рук и головы, и подмножество таких движений присутствует в «жонглировании мячами». Более того, интересно отметить, что первые два кадра показывают какой-то финт ногами с футбольным мячом, и он также определяется как жонглирование футбольным мячом.

3.4. СЕМАНТИЧЕСКАЯ СЕГМЕНТАЦИЯ С ИСПОЛЬЗОВАНИЕМ СЛАБОЙ РАЗМЕТКИ

Семантическая сегментация – это задача, в которой по входному изображению нам нужно обучить модель, способную предсказать категорию каждого пикселя в изображении. В современных методах (Chen et al., 2016; Zhao et al., 2017a) модель обычно представляет собой сверточные нейронные сети, которые обучаются с использованием аннотаций на уровне пикселей. Однако модель сегментации, обученная на одном наборе данных, может плохо обобщаться на изображения из другого набора, из-за несовпадения домена (предметной области) между наборами. Таким образом, модель необходимо адаптировать к изображениям из целевого домена, в котором она будет работать. Но поскольку разметка целевых изображений может оказаться дорогостоящей или вовсе невозможной, нужно найти способ адаптировать обученную модель к целевому домену с минимальными затратами на разметку или даже без них.

Методы *адаптации домена без учителя* (unsupervised domain adaptation, UDA) для семантической сегментации были разработаны для решения проблемы несовпадения доменов без затрат на аннотирование целевых изображений. Методы, описанные в литературе, направлены на адаптацию модели, обученной в исходном домене с попиксельными эталонными метками, например из симулятора, который требует наименьших усилий по разметке,

к целевому домену, который не имеет какой-либо разметки. Эти описанные в литературе методы UDA для семантической сегментации разрабатываются в основном с использованием двух механизмов: *самообучения с псевдоразметкой* (pseudo-label self-training) и выравнивания распределения между исходным и целевым доменами. Для первого механизма попиксельная псевдоразметка генерируется с помощью таких стратегий, как *степень уверенности* (confidence score, Li et al., 2019; Hung et al., 2018) или *обучение в режиме самоуправления* (self-paced learning, Zou et al., 2018), но подобная псевдоразметка специфична для целевого домена и не учитывает сопоставление между доменами. В случае второго механизма можно рассматривать различные пространства для работы процедуры сопоставления, такие как пиксель (Hoffman et al., 2018; Murez et al., 2018), признак (Hoffman et al., 2016; Zhang et al., 2017), выход (Tsai et al., 2018; Chen et al., 2018) и патч (Tsai et al., 2019). Однако сопоставление, выполняемое этими методами, не зависит от категории, что может быть проблематично, поскольку разрыв между доменами может варьироваться в зависимости от категории.

Проблема отсутствия разметки в целевом домене может быть решена путем введения концепции использования *слабой разметки* (weak labels) в целевом наборе данных для адаптации. Такую слабую разметку можно использовать для сопоставления по категориям между исходным и целевым доменами, а также для обеспечения соблюдения ограничений на категории, присутствующие в изображении. Может быть несколько форм слабой разметки – метки на уровне изображения, точечные метки, которые мы исследуем в этом тексте, а также другие формы слабой разметки, такие как плотность пикселей определенных категорий, количество объектов и т. д., которые довольно легко приобрести у аннотатора. Важно отметить, что наша слабая разметка может быть получена на основе предсказания модели в условиях UDA или предоставлена человеком-оракулом в парадигме *адаптации домена со слабым обучением* (weakly-supervised domain adaptation, WDA), как показано на рис. 3.9. Стоимость целевой разметки в UDA равна нулю, а в случае WDA очень мала, как мы увидим далее.

Литература по адаптации домена для моделей семантической сегментации посвящена только методам обучения без учителя (UDA), и ее можно разделить на три категории: адаптация на уровне пикселей (Hoffman et al., 2018; Murez et al., 2018; Wu et al., 2018), которая направлена на сопоставление пространства входного изображения; обучение псевдометкам (Zou et al., 2018; Sadat Saleh et al., 2018; Lian et al., 2019), которое направлено на маркировку немеченых целевых данных, использование исходной модели и ее использование для адаптации, а также адаптация признаков или выходного пространства (Tsai et al., 2018; Chen et al., 2018; Tsai et al., 2019; Du et al., 2019), целью которой является согласование выходного пространства между исходным и целевым доменами. Эффективность этих методов довольно низка по сравнению с методами с сильным обучением. Далее мы представляем наш метод (Paul et al., 2020), который можно использовать как для адаптации без разметки, так и для адаптации со слабой разметкой, а еще для преодоления разрыва в качестве с минимальными затратами на аннотации или без них. Мы начнем с формального определения задачи.



Рис. 3.9 ❖ На этом рисунке показано, как мы можем использовать слабую разметку на уровне изображения для адаптации домена двумя различными способами – либо вычисляемыми, т.е. псевдослабыми метками (адаптация домена без учителя, UDA), либо получаемыми от человеческого оракула (доменная адаптация со слабым обучением, WDA)

Определение задачи. У нас есть два домена – исходный и целевой. Задача заключается в том, чтобы адаптировать к целевому домену модель сегментации, обученную в исходном домене. В исходном домене мы имеем изображения и пиксельные метки, обозначаемые как $\mathcal{J}_s = \{X_s^i, Y_{sf}^{iN_s}\}_{i=1}^{N_s}$. Наш целевой набор данных содержит изображения и метки только на уровне изображения, такие как $\mathcal{J}_t = \{X_t^i, Y_t^{iN_t}\}_{i=1}^{N_t}$. Векторы $X_s, X_t \in \mathbb{R}^{H \times W \times 3}$, $Y_s \in \mathbb{R}^{H \times W \times C}$ являются пиксельными *уни-тарными* (one-hot) векторами, $y_t = \mathbb{R}^C$ – многопозиционный (multihot) вектор, представляющий категорию изображения, а C – количество категорий как для исходного, так и для целевого наборов данных. Напомним, что в одном унитарном векторе только один из элементов вектора равен единице, а остальные равны нулю, в то время как в многопозиционных векторах несколько элементов вектора могут быть равны единице. Такие метки на уровне изображения y_t называются *слабыми метками*, так как они представляют собой гораздо более слабую форму разметки по сравнению с попиксельной разметкой. Мы можем либо найти их, и в этом случае мы называем их *псевдослабыми метками* (pseudo-weak labels, метод UDA), либо получить их от человеческого оракула и назвать их *слабыми метками оракула* (oracle-weak labels, метод WDA). Получение слабых меток мы обсудим в разделе 3.4.4. Исходя из начальных условий, задача состоит в том, чтобы адаптировать модель сегментации G , обученную на исходном наборе данных \mathcal{J}_s , к целевому набору данных \mathcal{J}_t .

3.4.1. Слабые метки для классификации категорий

Мы используем слабые метки y_t и учим модель предсказывать наличие/отсутствие категорий в целевых изображениях. Сначала пропускаем целевые

изображения X_t через \mathbf{G} , чтобы получить прогнозы $A_t \in \mathbb{R}^{H \times W \times C}$, а затем применяем глобальный пулинговый слой для получения единого вектора прогнозов для каждой категории:

$$p_t^c = \sigma_s \left[\frac{1}{k} \log \frac{1}{H'W'} \sum_{h',w'} \exp k A_t^{(h',w',c)} \right], \quad (3.11)$$

где σ_s – сигмовидная функция, такая, что p_t представляет вероятность появления конкретной категории на изображении. Обратите внимание, что уравнение (3.11) представляет собой гладкую аппроксимацию функции \max . Чем выше значение k , тем лучше оно приближается к \max . Мы принимаем $k = 1$, поскольку не хотим, чтобы сеть фокусировалась только на максимальном значении прогноза, которое может быть зашумленным, но также и на других прогнозах, которые могут иметь высокие значения. Используя p_t и слабые метки y_t , мы можем вычислить бинарную кросс-энтропийную потерю по категориям:

$$\mathcal{L}_c(X_t; \mathbf{G}) = \sum_{c=1}^C -y_t^c \log(p_t^c) - (1 - y_t^c) \log(1 - p_t^c). \quad (3.12)$$

Эта часть процесса изображена в нижнем потоке рис. 3.10. Данная функция потерь \mathcal{L}_c помогает идентифицировать категории, которые отсутствуют/присутствуют на конкретном изображении, и заставляет сеть сегментации \mathbf{G} обращать внимание на те объекты/вещи, которые частично идентифицируются, когда исходная модель используется непосредственно на целевых изображениях.

3.4.2. Слабые метки для выравнивания признаков

Потеря классификации с использованием слабых меток, введенная в (3.12), упорядочивает сеть, фокусируясь на определенных категориях. Однако сопоставление распределения по исходному и целевому доменам пока не рассматривается. Как обсуждалось в предыдущем разделе, представленные в литературе методы сопоставляют между доменами либо пространство признаков (Hoffman et al., 2016), либо выходное пространство (Tsai et al., 2018). Однако такое сопоставление не зависит от категории, поэтому оно может сопоставлять признаки категорий, которых нет в определенных изображениях. Более того, признаки, принадлежащие к разным категориям, могут иметь разные разрывы в домене. Таким образом, сопоставление по категориям может быть полезным, но оно не было широко изучено в UDA в части семантической сегментации. Чтобы решить эти проблемы, мы используем слабые метки на уровне изображения для сопоставления по категориям в пространстве признаков. В частности, мы получаем категориальные признаки для каждого изображения с помощью карты внимания, то есть предсказания сегментации, руководствуясь нашим модулем классификации с использованием слабых меток, а затем сопоставляем эти признаки между исходным

и целевым доменами. Далее обсудим механизм пулинга категориальных признаков, а затем метод *состязательного сопоставления* (adversarial alignment).

Объединение признаков по категориям. Пусть у нас есть признаки последнего слоя $F \in \mathbb{R}^{H' \times W' \times 2048}$ и предсказание сегментации $A \in \mathbb{R}^{H' \times W' \times C}$. Отсюда мы получаем категориальные признаки для s -й категории в виде 2048-мерного вектора, используя предсказание как внимание, обращенное на следующие признаки:

$$\mathcal{F}^c = \sum_{h', w'} \sigma(A)^{(h', w', c)} F^{(h', w')}, \quad (3.13)$$

где $\sigma(A)$ – тензор размерности $H' \times W' \times C$, каждый канал которого в измерении категории представляет внимание по категориям, полученное softmax-операцией σ над пространственными измерениями. В итоге $\sigma(A)^{(h', w', c)}$ – это скаляр, $F^{(h', w')}$ представляет собой 2048-мерный вектор, а \mathcal{F}^c – это суммарный признак $F^{(h', w')}$, взвешенный по $\sigma(A)^{(h', w', c)}$ на пространственной карте $H' \times W'$. Обратите внимание, что мы опускаем индексы s, t для источника и цели, поскольку используем одну и ту же операцию для получения категориальных признаков для обоих доменов. Далее мы представляем механизм для согласования этих признаков между доменами. Обратите внимание, что мы будем использовать \mathcal{F}^c для обозначения пула признаков для категории s и \mathcal{F}^c для обозначения множества пулов признаков для всех категорий. Пулинг признаков по категориям показан в середине рис. 3.10.

Сопоставление категориальных признаков. Для обучения сети сегментации G таким образом, чтобы были сопоставлены исходные и целевые категориальные признаки, мы используем состязательную потерю при применении дискриминаторов для конкретных категорий $\mathbf{D}^c = \{\mathbf{D}^c\}_{c=1}^C$. Причина использования дискриминаторов для конкретных категорий заключается в том, чтобы гарантировать, что распределение признаков для каждой категории может быть сопоставлено независимо, что позволяет избежать зашумленного моделирования распределения из смеси категорий. На практике мы обучаем C различных дискриминаторов, специфичных для категорий, чтобы различать категориальные признаки, взятые из исходного и целевого изображений. Функция потерь для обучения дискриминаторов \mathbf{D}^c выглядит следующим образом:

$$\mathcal{L}_d^c(\mathcal{F}_s^c, \mathcal{F}_t^c, \mathbf{D}^c) = \sum_{c=1}^C -y_s^c \log \mathbf{D}^c(\mathcal{F}_s^c) - y_t^c \log(1 - \mathbf{D}^c(\mathcal{F}_t^c)). \quad (3.14)$$

Стоит отметить, что при обучении дискриминаторов мы вычисляем потери только для тех категорий, которые присутствуют в конкретном изображении, – с помощью слабых меток $y_s, y_t \in \mathbb{B}^C$, которые указывают, встречается категория на изображении или нет. Тогда состязательную потерю для целевых изображений при обучении сети сегментации G можно выразить следующим образом:

$$\mathcal{L}_{adv}^C(\mathcal{F}_t^C; \mathbf{G}, \mathbf{D}^C) = \sum_{c=1}^C -y_t^c \log \mathbf{D}^c(\mathcal{F}_t^c). \quad (3.15)$$

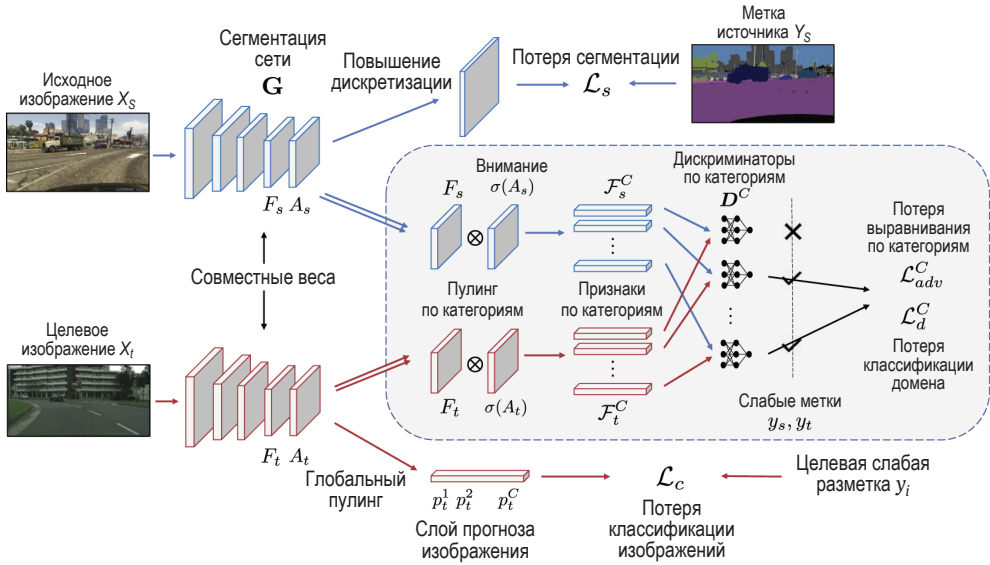


Рис. 3.10 ❖ Предлагаемая архитектура состоит из сети сегментации \mathbf{G} и модуля слабой разметки. Мы вычисляем попиксельные потери сегментации \mathcal{L}_s для исходных изображений и потери классификации изображений \mathcal{L}_c , используя слабые метки y_t для целевых изображений. Обратите внимание, что слабые метки могут быть найдены как псевдослабые метки или предоставлены человеком-ораклом. Затем мы используем выходной прогноз A , преобразуем его в карту внимания $\sigma(A)$ и объединяем признаки по категориям \mathcal{F}^C . Далее эти признаки сопоставляются между исходным и целевым доменами с использованием потери сопоставления по категориям \mathcal{L}_{adv}^C , управляемой дискриминаторами по категориям \mathbf{D}^C , которые обучены с помощью потерь классификации доменов \mathcal{L}_d^C .

Точно так же мы используем целевые слабые метки y_t , чтобы сопоставить только те категории, которые присутствуют в целевом изображении. Минимизируя \mathcal{L}_{adv}^C , сеть сегментации пытается обмануть дискриминатор, максимизируя вероятность того, что целевой категориальный признак будет рассматриваться как взятый из исходного распределения. Эти функции потерь в (3.14) и (3.15) получены в правой части среднего прямоугольника на рис. 3.10.

3.4.3. Оптимизация сети

Тренировка дискриминатора. Мы обучаем множество из C различных дискриминаторов для каждой категории c . Мы используем исходное и целевое изображения для обучения дискриминаторов, которые учатся различать ка-

тегориальные признаки, взятые либо из исходного, либо из целевого домена. Задача оптимизации для обучения дискриминатора может быть формально выражена следующим образом: $\min_{D^C} \mathcal{L}_d^C(\mathcal{F}_s^C, \mathcal{F}_t^C)$. Заметим, что каждый дискриминатор обучается только с признаками, подвергнутыми пулингу в соответствии с его конкретной категорией. Поэтому для заданного изображения мы обновляем только те дискриминаторы, которые соответствуют категориям, присутствующим на изображении.

Обучение сети сегментации. Мы обучаем сеть сегментации с попиксельной кросс-энтропийной потерей \mathcal{L}_s на исходных изображениях, потерей классификации изображений \mathcal{L}_c и состязательной потерей \mathcal{L}_{adv} на целевых изображениях. Чтобы обучить \mathbf{G} , объединяем эти функции потерь следующим образом:

$$\min_{\mathbf{G}} L_s(X_s) + \lambda_c L_c(X_t) + \lambda_d L_{adv}^C(F_t^C). \quad (3.16)$$

Мы следуем стандартной процедуре обучения GAN (Goodfellow et al., 2014), чтобы поочередно обновлять \mathbf{G} и \mathbf{D}^C . Обратите внимание, что при вычислении \mathcal{L}_{adv}^C используются дискриминаторы по категориям \mathbf{D}^C . Поэтому мы исправляем \mathbf{D}^C и градиенты обратного распространения только для сети сегментации \mathbf{G} .

3.4.4. Получение слабой разметки

Выше мы предложили механизм использования слабых меток на уровне изображения для целевых изображений и адаптации модели сегментации между исходным и целевым доменами. В этом разделе мы объясняем два метода получения такой слабой разметки на уровне изображения.

Псевдослабые метки (UDA). Одним из способов получения слабых меток является их непосредственное нахождение с использованием имеющихся у нас данных, т. е. исходных изображений/меток и целевых изображений, что относится к случаю адаптации домена без учителя (UDA). В этой работе мы используем базовую модель (Tsai et al., 2018), чтобы адаптировать модель, обученную на эталонных данных, к целевому домену, а затем получаем слабые метки целевых изображений следующим образом:

$$y_t^c = \begin{cases} 1, & \text{если } p_t^c > T \\ 0, & \text{если не так} \end{cases}, \quad (3.17)$$

где p_t^c – вероятность для категории c , вычисленная согласно (3.11), а T – порог, который мы установили равным 0,2 во всех экспериментах, если не указано иное. На практике мы вычисляем слабые метки онлайн во время обучения и избегаем любого дополнительного шага вывода. В частности, мы пропускаем через сеть целевое изображение, получаем слабые метки, используя (3.17), а затем вычисляем функции потерь в (3.16). Поскольку слабые метки,

полученные таким образом, не требуют участия человека, адаптация с использованием таких меток является адаптацией без учителя.

Адаптация домена со слабым обучением (WDA). В этой форме мы получаем слабые метки, запрашивая у человека-оракула список категорий, которые встречаются в целевом изображении. Поскольку мы нуждаемся в работе оракула над целевыми изображениями, то называем это адаптацией домена со слабым обучением (WDA). Стоит отметить, что подход WDA может быть практически полезным, поскольку получать такие слабые метки, присвоенные человеком, намного проще, чем попиксельную разметку. Кроме того, ранее не проводилось никаких исследований с использованием этого подхода для адаптации домена.

Чтобы показать, что наш метод может использовать различные формы слабых меток, полученных от оракула, мы дополнительно вводим *точечное обучение*¹ (point supervision) согласно Bearman et al. (2016), что лишь незначительно увеличивает усилия по сравнению с разметкой на уровне изображения. В этом сценарии мы случайным образом получаем одну координату пикселя каждой категории, принадлежащей изображению, т. е. множество кортежей $\{(h^c, w^c, c) | \forall y_i^c = 1\}$. Для изображения мы вычисляем потери следующим образом: $\mathcal{L}_{point} = -\sum_{\forall y_i^c=1} y_i^c \log(O_i^{(h^c, w^c, c)})$, где $O_i \in \mathbb{R}^{H \times W \times C}$ – выходное предсказание цели после попиксельной операции softmax.

3.4.5. Применения

В этом разделе мы проводим оценку нашего фреймворка адаптации предметной области для семантической сегментации, где источником является набор данных, состоящий из смоделированных изображений уличных сцен (GTA5; Richter et al., 2016), а целевой набор состоит из изображений реального мира (городские пейзажи; Cordts et al., 2016). В качестве метрики мы используем отношение IoU. Для сети сегментации G применяем фреймворк DeepLab-v2 (Chen et al., 2016) с архитектурой ResNet-101 (He et al., 2016). Мы извлекаем признаки F_s, F_t перед слоем *расширенного пространственно-пирамидального пулинга* (atrous spatial pyramid pooling, ASPP). В качестве категориальных дискриминаторов $D^C = \{D^c\}_{c=1}^C$ используем C отдельных сетей, каждая из которых состоит из трех полносвязных слоев, имеющих $\{2048, 2048, 1\}$ узлов с активацией ReLU. На рис. 3.11 даны результаты, представленные в публикациях, и результаты применения нашего метода для различных используемых нами аннотаций.

¹ Точечное обучение подразумевает указание на интересующие объекты на изображении с помощью щелчков мыши. Точечное обучение значительно дешевле и занимает меньше времени по сравнению с традиционными методами сильной разметки, такими как рисование ограничительной рамки и попиксельная маркировка изображений. – Прим. перев.

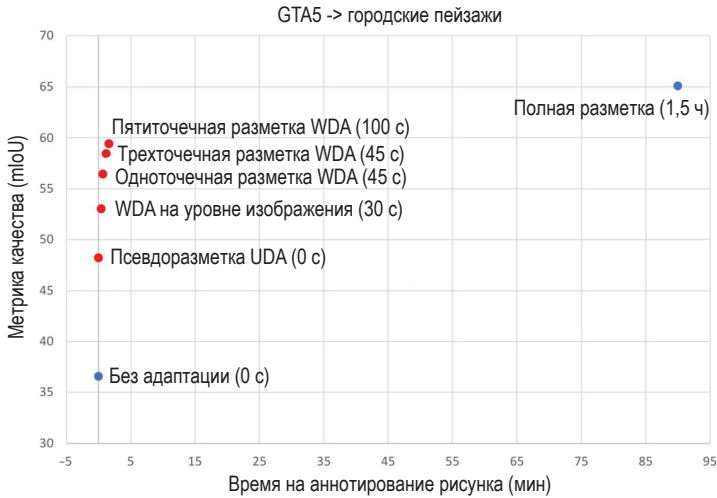


Рис. 3.11 ❖ Сравнение результатов в сценарии «GTA5 → городские пейзажи» с различными уровнями разметки: без целевых меток («Нет адаптации» и «UDA»), слабые метки на уровне изображений (30 с), одноточечные метки (45 с) и полная попиксельная разметка («Все помечены»), которая занимает 1,5 часа на изображение в соответствии с (Cordts et al., 2016)

Современные методы

Анализируя современные методы, представленные в литературе, мы видим, что все они не требуют затрат на разметку на целевой стороне. В такой обстановке достигаемые уровни качества относительно невысоки. Например, в упомянутом выше сценарии с GTA 5 в качестве источника и набором Cityscapes (городские пейзажи) в качестве цели показатель качества современных методов составляет 45,4 (Chang et al., 2019), 46,5 (Tsai et al., 2019). al., 2019) и 47,2 (Li et al., 2019). Стоит отметить, что эти показатели выше, чем у модели, обученной на источнике и примененной непосредственно к цели, которая показала точность всего 36,6 %. Однако это намного ниже, чем значение 64,4, полученное в сценарии, когда все целевые изображения помечены аннотациями на уровне пикселей, но в таком случае разметка одного изображения занимает около 90 мин.

Адаптация домена без учителя (UDA)

По сравнению с методами, описанными в литературе, когда мы используем базовый метод (Tsai et al., 2018) для сопоставления только выходного пространства, мы получаем уровень качества 41,4. Затем, когда используем потерю классификации согласно уравнению (3.12), получаем качество всего 46,7. Когда мы добавляем потерю сопоставления по категориям согласно уравнению (3.15), то получаем качество 48,2. Отметим, что в этом методе мы используем псевдослабые метки, которые находит сама сеть по уравнению (3.17).

Адаптация домена со слабым обучением (WDA)

Мы используем два разных типа аннотаций в WDA, о них пойдет речь ниже. Обратите внимание, что другими формами слабого обучения могут быть подсчет объектов, очень грубая оценка области охвата категорий, грубое начертание (scribble) и т. д.

Обучающая разметка на уровне изображения. Когда мы используем аннотации на уровне изображения, полученные от пользователя, и обучаем сеть сегментации, используя только потери классификации в уравнении (3.12), мы получаем качество 52. Когда добавляем потерю сопоставления домена по категориям, то получаем качество 53. Следует отметить, что нет опубликованных работ, в которых использовались бы слабые метки людей-оракулов для выполнения WDA. Что касается результатов, интересно отметить, что значительное повышение качества при использовании WDA по сравнению с UDA наблюдается для таких категорий, как грузовик, автобус, поезд и мотоцикл. Одна из причин заключается в том, что эти категории наиболее недопредставлены как в исходных, так и в целевых наборах данных. Таким образом, они не предсказываются в большинстве целевых изображений, но использование слабых меток помогает лучше их идентифицировать.

Точечная обучающая разметка. На наш взгляд, заслуживает внимания еще один метод точечной разметки (Bearman et al., 2016), который лишь незначительно увеличивает время аннотирования по сравнению с разметкой на уровне изображения. Мы воспользовались этим методом и случайным образом выбираем один пиксель для каждой категории в каждом целевом изображении в качестве обучающего. В этом случае все детали и модули во время обучения одинаковы. С одной точкой, помеченной для каждой категории на каждом изображении, мы получаем показатель качества 56,4. Когда мы увеличиваем количество аннотаций до трех или пяти точек, помеченных для каждой категории на каждом изображении, показатель возрастает до 58,4 и 59,4 соответственно. Стоит отметить, что стоимость разметки у данного метода довольно низкая, как показано на рис. 3.11, но тем не менее он может обеспечить качество, сравнимое с попиксельной разметкой, которая требует огромных затрат.

3.4.6. Визуализация выходного пространства

Некоторые визуализации вероятности предсказания сегментации для каждой категории изображены на рис. 3.12. До использования каких-либо слабых меток (третий ряд) вероятность может быть низкой, даже если на этом изображении присутствует категория. Однако, основываясь на этих первоначальных прогнозах, наша модель может оценить категории, а затем явно указать их наличие/отсутствие в предлагаемых потерях классификации и потерях сопоставления. Четвертая строка на рис. 3.12 показывает, что такие псевдослабые метки помогают сети обнаруживать области объекта/содержимого для лучшей сегментации. Например, четвертый и пятый столбцы

показывают, что хотя исходные вероятности предсказания довольно низки, результаты с использованием псевдослабых меток оцениваются правильно. Более того, последняя строка показывает, что прогнозы могут быть улучшены, когда у нас есть слабые метки, предоставленные оракулом.

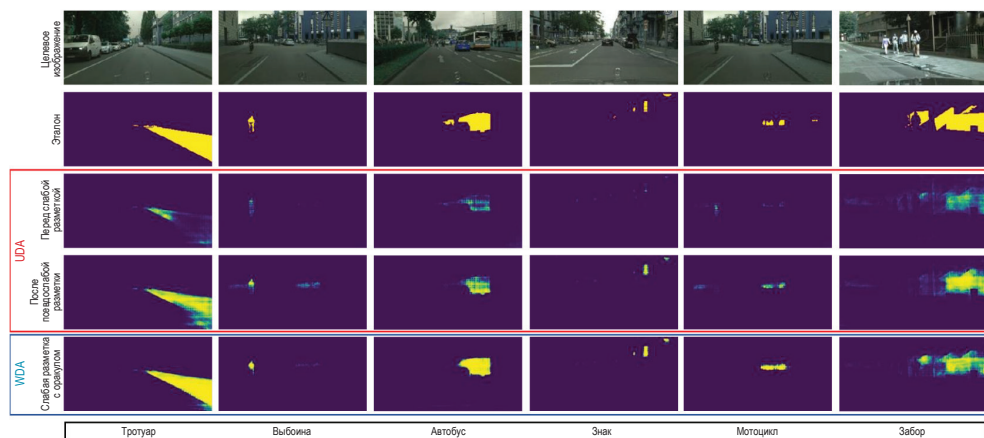


Рис. 3.12 ❖ Визуализация вероятности предсказания сегментации по категориям до и после использования псевдослабых меток в GTA5 – Cityscapes. До адаптации сеть лишь частично выделяет области с малой вероятностью, а использование псевдослабых меток помогает адаптированной модели получать гораздо более качественные сегменты и приближается к модели с использованием слабой разметки с оракулом

3.5. ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ СО СЛАБОЙ РАЗМЕТКОЙ ДЛЯ ДИНАМИЧЕСКИХ ЗАДАЧ

До сих пор в этой главе мы рассматривали в свете ограниченной разметки в основном статические задачи. Однако в реальных сценариях нам, возможно, придется построить систему, которая взаимодействует с окружающей средой для выполнения задачи и, таким образом, должна быть динамичной по своей природе, поскольку от ее текущего действия зависит точка данных, которую она получит следующей. Подобно статическим задачам, таким как классификация, локализация и сегментация, изучение динамических задач также довольно сложно по своей природе. *Обучение с подкреплением* (reinforcement learning, RL) с использованием глубоких нейронных сетей (deep neural networks, DNN) продемонстрировало огромные успехи в таких динамических задачах, как игры (Mnih et al., 2015; Silver et al., 2016), в решении сложных задач робототехники (Levine et al., 2016; Duan et al., 2016) и т. д. Однако из-за скудного подкрепления эти алгоритмы часто требуют огромного количества взаимодействий с окружающей средой, что дорого обходится в реальных приложениях, таких как беспилотные автомобили (Bojarski et

al., 2016) и манипуляции с использованием реальных роботов (Levine et al., 2016). Созданные вручную *плотные функции вознаграждения* (dense reward function) могут смягчить эти проблемы; однако в целом сложно разработать подробные функции вознаграждения для сложных реальных задач.

Для более быстрого изучения политик потенциально может применяться *имитационное обучение* (imitation learning, IL) с использованием демонстрационных примеров, созданных экспертом (Argall et al., 2009). Но качество работы алгоритмов IL (Ross et al., 2011) зависит не только от компетентности эксперта, предоставляющего примеры, но и от распределения в пространстве состояний, представленного образцами, особенно в случае состояний высокой размерности. Во избежание такой зависимости от эксперта некоторые исследователи (Sun et al., 2017; Cheng et al., 2018) идут по пути совмещения RL и IL. Однако эти методы предполагают использование экспертной оценки, что может оказаться непрактичным в реальных сценариях.

Учитывая недостатки методов, предложенных другими исследователями, мы представляем стратегию, которая начинается с IL, а затем переключается на RL (Paul et al., 2019). На этапе IL наш фреймворк выполняет предварительное обучение с учителем, целью которого является нахождение политики, которая лучше всего описывает демонстрационные примеры экспертов. Однако из-за ограниченной доступности экспертных примеров политика, найденная с помощью IL, будет содержать ошибки, которые затем можно устранить с помощью RL. Но обратите внимание, что функция вознаграждения в RL все еще скудна, что затрудняет обучение. Поэтому мы представляем метод, который использует подготовленные человеком примеры, чтобы разделить всю задачу на более мелкие *подцели* (промежуточные цели) и использовать их в качестве функции вознаграждения на этапе RL.

Имея набор примеров, люди могут быстро определить путевые точки, через которые необходимо пройти для достижения цели. Мы склонны разбивать сложную задачу на подцели и пытаться достичь их в наилучшем возможном порядке. Присущее людям *предварительное знание* помогает решать задачи намного быстрее (Andreas et al., 2017; Dubey et al., 2018), чем использование только обучающих примеров. Человеческая психология «разделяй и властвуй» чрезвычайно эффективна при решении различных задач, и она послужила основой для нашего алгоритма, который учится разделять пространство состояний на подцели, используя экспертные примеры. Выученные подцели обеспечивают дискретный сигнал вознаграждения, в отличие от непрерывного вознаграждения, основанного на ценности (Ng et al., 1999; Sun et al., 2018), которое может быть ошибочным, особенно при ограниченном количестве примеров в задачах с отдаленной целью. Поскольку набор экспертных примеров может не содержать всех состояний, которые агент может принять во время обучения на этапе RL, мы расширяем предиктор подцели с помощью классификации одного класса, чтобы иметь возможность работать с такими недопредставленными состояниями. Мы проводим эксперименты над тремя целевыми задачами на наборе MuJoCo (Тодоров, 2014) с разреженным вознаграждением только за успешное окончание, которые современные модели RL, IL или их комбинации не могут решить.

Постановка задачи. Рассмотрим стандартную схему RL, в которой агент взаимодействует со средой, которую можно смоделировать с помощью *марковского процесса принятия решений* (markov decision process, MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \mathcal{P}_0)$, где \mathcal{S} – множество состояний, \mathcal{A} – множество действий, r – скалярная функция вознаграждения, $\gamma \in [0, 1]$ – коэффициент дисконтирования, а \mathcal{P}_0 – начальное распределение состояний. Наша цель – выучить политику $\pi_\theta(\mathbf{a}|\mathbf{s})$, где $\mathbf{a} \in \mathcal{A}$, оптимизирующую ожидаемое дисконтированное вознаграждение $\mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$, где $\tau = (\dots, \mathbf{s}_t, \mathbf{a}_t, r_t, \dots)$ и $\mathbf{s}_0 \sim \mathcal{P}_0$, $\mathbf{a}_t \sim \pi_\theta(\mathbf{a}|\mathbf{s}_t)$ и $\mathbf{s}_{t+1} \sim \mathcal{P}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.

При разреженном вознаграждении оптимизация ожидаемого дисконтированного вознаграждения со скидкой с использованием RL может вызвать затруднения. В таких случаях может быть полезно использовать множество примеров состояния-действия $\mathcal{D} = \{(\mathbf{s}_{ti}, \mathbf{a}_{ti}^*)\}_{t=1}^{n_i}$, сгенерированных экспертом, для управления процессом обучения. Здесь n_d – количество примеров в наборе данных, а n_i – длина i -го примера. Мы предлагаем методику эффективного использования \mathcal{D} путем обнаружения подцелей в этих примерах и использования их для разработки внешней функции вознаграждения.

Определение подцели

Пусть пространство состояний \mathcal{S} разделено на множества состояний как $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{n_g}\}$, таких, что $\mathcal{S} = \bigcup_{i=1}^{n_g} \mathcal{S}_i$ и $\bigcap_{i=1}^{n_g} \mathcal{S}_i = \emptyset$, где n_g – количество подцелей, указанных пользователем. Для каждого $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ мы полагаем, что конкретное действие переводит агента от одной подцели к другой тогда и только тогда, когда $\mathbf{s} \in \mathcal{S}_i$, $\mathbf{s}' \in \mathcal{S}_j$ для некоторых $i, j \in \mathbf{G} = \{1, 2, \dots, n_g\}$ и $i \neq j$.

Предположим, что существует порядок, в котором группы состояний появляются в примерах, как показано на рис. 3.13(b). Однако состояния внутри этих групп состояний могут появляться в примерах в произвольном порядке. Эти группы состояний не определены априори, и наш алгоритм нацелен на оценку данных разделов. Надо сказать, что такое упорядочение является естественным в некоторых реальных сценариях, где определенная подцель

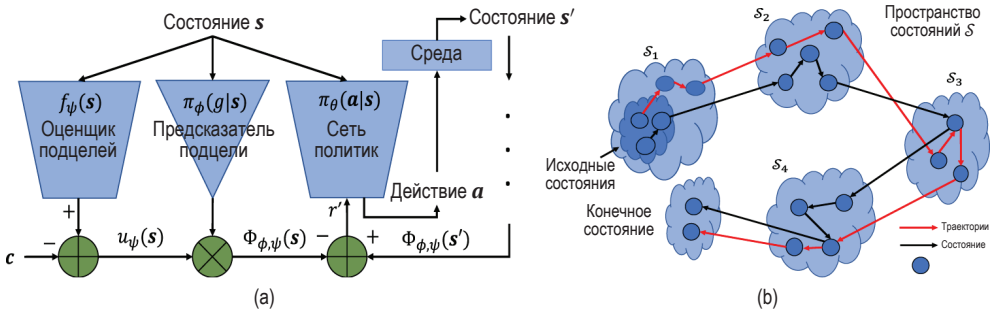


Рис. 3.13 ❖ (а) Здесь показан обзор предложенного нами фреймворка для обучения сети политик вместе с функцией вознаграждения на основе расширенных внешних подцелей; (б) пример разделения состояния с двумя независимыми примерами черного и красного цветов. Обратите внимание, что окончательное состояние (terminal state) показано как отдельный раздел состояний, потому что мы предполагаем, что оно указано средой, а не выучено

может быть достигнута только после достижения одной или нескольких предыдущих подцелей. Можно считать, что состояния в примере появляются в порядке возрастания индексов подцелей, т. е. достижение подцели j сложнее, чем достижение подцели i ($i < j$). Это дает нам естественный способ определения внешней функции вознаграждения, которая поможет ускорить поиск политики. Кроме того, все примеры должны начинаться с начального распределения состояний и заканчиваться конечными состояниями.

3.5.1. Обучение прогнозированию подцелей

Мы используем набор данных \mathcal{D} для разделения пространства состояний на n_g подцелей, где n_g является гиперпараметром. Мы учим нейронную сеть аппроксимировать $\pi_\phi(g|s)$, которая при заданном состоянии $s \in \mathcal{S}$ предсказывает функцию распределения вероятностей по возможным разбиениям подцелей $g \in \mathbf{G}$. Порядок, в котором подцели встречаются в демоверсии, т. е. $\mathcal{S}_1 < \mathcal{S}_2 < \dots \mathcal{S}_{n_g}$, которое можно вывести из нашего предположения, упомянутого выше, действует как контролирующий сигнал. Фреймворк для обучения $\pi_\phi(g|s)$ является итеративным и подразумевает чередование между этапом обучения и этапом вывода/коррекции, как объясняется далее.

Этап обучения. На этом этапе мы считаем, что у нас есть набор кортежей (s, g) , которые мы используем для обучения функции π_ϕ . Это можно представить как задачу мультиклассовой классификации с n_g категориями. Мы оптимизируем следующую кросс-энтропийную функцию потерь:

$$\pi_\phi^* = \operatorname{argmin}_{\pi_\phi} \frac{1}{N} \sum_{i=1}^{n_d} \sum_{t=1}^{n_i} \sum_{k=1}^{n_g} -\mathbf{1}\{g_{ti} = k\} \log \pi_\phi(g = k|s_{ti}), \quad (3.18)$$

где $\mathbf{1}$ – индикаторная функция, а N – количество состояний в наборе данных \mathcal{D} . Начнем с того, что у нас нет никаких меток g , поэтому мы рассматриваем равномерное распределение всех подцелей в \mathbf{G} по каждому примеру. То есть если дан пример состояний $\{s_{1i}, s_{2i}, \dots, s_{n_i i}\}$ для некоторого $i \in \{1, 2, \dots, n_d\}$, то начальные подцели равного разделения будут такими:

$$g_{ti} = j, \quad \forall \left\lfloor \frac{(j-1)}{n_g} n_i \right\rfloor < t \leq \left\lfloor \frac{j}{n_g} n_i \right\rfloor, \quad j \in \mathbf{G}. \quad (3.19)$$

С такой начальной схемой разметки аналогичные состояния в примерах могут иметь разные метки, но ожидается, что сеть будет сходиться при оценке максимального правдоподобия (maximum likelihood estimate, MLE) всего набора данных. Мы также оптимизируем CASL (Paul et al., 2018), представленный в разделе 3.3.3, для стабильного обучения, поскольку начальные метки могут быть ошибочными. На следующей итерации этапа обучения мы используем прогнозные метки подцелей, которые получаем следующим образом.

Шаг вывода. Хотя метки равномерного разделения в уравнении (3.19) могут иметь сходные состояния в разных примерах, сопоставленные с несовпадающими

подцелями, обученная сеть, моделирующая π_ϕ , сопоставляет сходные состояния с одной и той же подцелью. Но уравнение (3.18) и, таким образом, предсказания π_ϕ не учитывают естественный временной порядок подцелей. Даже при использовании таких архитектур, как *рекуррентные нейронные сети* (recurrent neural network, RNN), может быть лучше явно наложить подобные ограничения временного порядка, чем полагаться на сеть для их изучения. Мы вводим такие ограничения порядка, используя *динамическое искажение времени* (dynamic time warping, DTW).

Формально для i -го примера в \mathcal{D} мы получаем следующее множество: $\{\mathbf{s}_{ti}, \pi_\phi(g|\mathbf{s}_{ti})\}_{t=1}^{n_i}$, где π_ϕ – вектор, представляющий PMF по подцелям G . Однако, поскольку предсказания не учитывают временной порядок, ограничение на предмет того, что подцель j возникает после подцели i при $i < j$, не сохраняется. Чтобы наложить такие ограничения, мы используем DTW между двумя последовательностями $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n_g}\}$, которые являются стандартными базисными векторами в n_g -мерном евклидовом пространстве, и $\{\pi_\phi(g|\mathbf{s}_{1i}), \pi_\phi(g|\mathbf{s}_{2i}), \dots, \pi_\phi(g|\mathbf{s}_{n_i i})\}$. В качестве меры подобия в DTW мы используем l_1 -норму разности двух векторов. В этом процессе мы связываем с каждым состоянием в примерах подцели, которые становятся новыми метками для обучения.

Затем проводим этап обучения, используя новые метки (вместо уравнения (3.19)), за которыми следует этап вывода, чтобы получить следующие метки подцели. Мы продолжаем этот процесс до тех пор, пока число меток подцелей, изменившихся между итерациями, не станет меньше определенного порога. Этот метод представлен в алгоритме 3.1, где верхний индекс k обозначает номер итерации в чередованиях обучения-вывода.

Вознаграждение с использованием подцелей. Порядок подцелей, как упоминалось ранее, обеспечивает естественный способ разработки функции вознаграждения следующим образом:

$$r'(\mathbf{s}, a, \mathbf{s}') = \gamma * \operatorname{argmax}_{j \in G} \pi_\phi(g = j|\mathbf{s}') - \operatorname{argmax}_{k \in G} \pi_\phi(g = k|\mathbf{s}), \quad (3.20)$$

где агент в состоянии \mathbf{s} выполняет действие a и достигает состояния \mathbf{s}' . Расширенная функция вознаграждения принимает вид $r + r'$. Учитывая, что у нас есть функция вида $\Phi_\phi(\mathbf{s}) = \operatorname{argmax}_{j \in G} \pi_\phi(g = j|\mathbf{s})$ и что $G = \{0, 1, \dots, n_{g-1}\}$ без

потери общности, так что для начального состояния $\Phi_\phi = (\mathbf{s}_0) = 0$, из работы (Ng et al., 1999) следует, что каждая оптимальная политика в $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r + r', \gamma, \mathcal{P}_0)$ будет также оптимальной в \mathcal{M} , исходном MDP. Однако новая функция вознаграждения способна помочь быстрее обучиться под нужную задачу.

Алгоритм 3.1. Обучение прогнозированию подцелей

Исходные данные: обучающий набор примеров \mathcal{D}

Выход: предиктор подцели $\pi_\phi(g|\mathbf{s})$

$k \leftarrow 0$

Получить g^k для каждого $\mathbf{s} \in \mathcal{D}$ с помощью уравнения (3.19)

повторить

Оптимизировать (3.18) чтобы получить π_ϕ^k

Предсказать PMF \mathbf{G} для каждого $\mathbf{s} \in \mathcal{D}$, используя π_ϕ^k
 Получить новые подцели g^{k+1} , используя PMF в DTW
 завершить = Да, если $|g^k - g^{k+1}| < \epsilon$, иначе Нет
 $k \leftarrow k + 1$

пока завершить \neq Да

3.5.2. Предварительное обучение с учителем

Как обсуждалось ранее, первоначальный способ использования обучающих примеров заключается в предварительном обучении сети политик π_θ с использованием обучающего набора в среде обучения с учителем. Мы предварительно обучаем сеть, оптимизируя следующий параметр:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n_d} \sum_{t=1}^{n_i} l(\pi_\theta(\mathbf{a}|\mathbf{s}_{ti}), \mathbf{a}_{ti}^*) + \lambda \|\theta\|_F^2, \quad (3.21)$$

где l – функция потерь, которая может быть кросс-энтропийной или регрессионной, в зависимости от дискретных или непрерывных действий. Обратите внимание, что непрерывные действия состоят из (μ, σ) , которые являются параметрами распределения Гаусса. Вторая часть уравнения (3.21) – потеря регуляризации l_2 . Политика, полученная после оптимизации уравнения (3.21), обладает способностью предпринимать действия с низким уровнем ошибок в состояниях, выбранных из распределения, индуцированного обучающим набором. Однако, как показано в работе Росса и др. (Ross et al., 2011), небольшая ошибка в начале будет увеличиваться квадратично от времени, поскольку агент начинает посещать состояния, которые не выбраны из распределения \mathcal{D} . Существуют такие алгоритмы, как DAGger, применяемые для точной настройки политики путем запроса экспертных решений в состояниях, посещенных после выполнения изученной политики. Однако такие запросы к эксперту часто обходятся очень дорого и в некоторых сценариях неосуществимы. Что еще более важно, поскольку DAGger стремится имитировать эксперта, в идеале он может только достичь его уровня качества, но не превзойти. По этой причине мы выполняем тонкую настройку политики, используя RL с внешней функцией вознаграждения, полученной после определения подцелей.

3.5.3. Практическое применение

В этом разделе мы представляем три сложные динамические задачи: используем нашу структуру для их решения и сравниваем их с другими современными методами, предложенными в различных публикациях.

Задачи. Мы проводим эксперименты в трех сложных средах, изображенных на рис. 3.14. Первая среда – это игра «Мяч в лабиринте» (BiMGame; van Baar et al., 2018), где задача состоит в том, чтобы переместить шарик из самого внешнего кольца в самое внутреннее, используя набор из пяти дискретных

действий – поворот по часовой стрелке и против часовой стрелки на 1° по двум главным измерениям доски и «пустая операция», где сохраняется текущая ориентация доски. Состояния представляют собой изображения размером 84×84 . Вторая среда – «Путешествие муравья» (AntTarget), в которой участвует муравей (Schulman et al., 2015). Задача состоит в том, чтобы достичь центра окружности радиусом 5 м, при этом муравей инициализируется на дуге окружности 45° . Состояние и действие непрерывны с 41 и 8 измерениями соответственно. В третьей среде – «Муравьиный лабиринт» (AntMaze) – используется тот же муравей, но путешествующий в U-образном лабиринте (Held et al., 2017). Муравей инициализируется на одном конце лабиринта, а целью является другой конец, обозначенный красным цветом на рис. 3.14с. Для всех задач мы используем разреженное вознаграждение только на основе завершающего состояния, то есть 1 только после достижения цели и 0 в противном случае. Стандартные методы RL, такие как A3C (Mnih et al., 2016), не могут решить эти задачи с таким скудным вознаграждением.

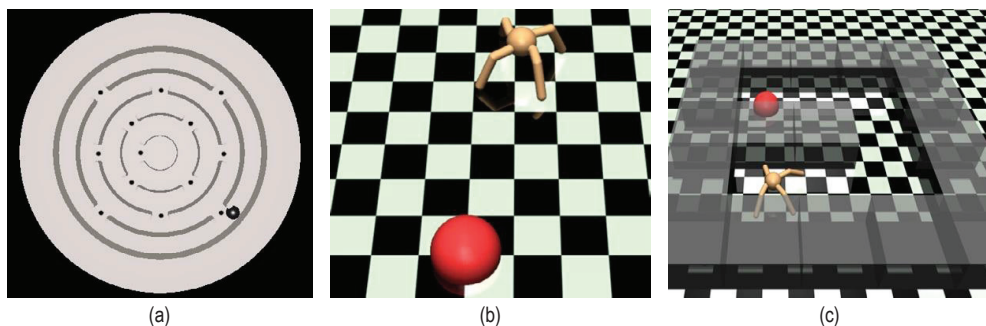


Рис. 3.14 ❖ На этом рисунке представлены три среды, которые мы используем: (a) игра «Мяч в лабиринте» (BiMGame); (b) передвижение муравья в открытой среде с конечной целью (AntTarget); (c) передвижение муравья в лабиринте с конечной целью (AntMaze)

Визуализация. Мы визуализируем подцели, обнаруженные нашим алгоритмом, и наносим их на плоскость x – y , как показано на рис. 3.15. Как видно в случае BiMGame, с 4 подцелями наш метод может обнаружить узкие места на доске как разные подцели. В случае AntTarget и AntMaze путь к цели более или менее поровну делится на подцели. Это показывает, что наш метод обнаружения подцелей может работать как в средах с узкими местами, так и без них.

Сравнение с другими методами. Мы в первую очередь сравниваем наш метод с другими методами RL, которые используют обучающие примеры или экспертную информацию – AggreVaTeD (Sun et al., 2017) и формирование вознаграждения на основе ценности (Ng et al., 1999), что эквивалентно $K = \infty$ в THOR (Sun et al., 2018). Сравнение представлено на рис. 3.16. Как можно заметить, ни один из исходных методов не показывает каких-либо признаков обучения задачам, за исключением ValueReward, который работает сравнимо

с предложенным нами методом только в случае AntTarget. Наш метод, с другой стороны, способен последовательно учиться и решать задачи в течение нескольких прогонов. На графиках прямыми линиями показаны совокупные вознаграждения экспертов, и методы имитации обучения, такие как DAgger (Ross et al., 2011), могут достичь только этой отметки. Наш метод способен превзойти эксперта во всех задачах. На самом деле для AntMaze даже с довольно неоптимальным экспертом (среднее накопительное вознаграждение всего 0,0002) наш алгоритм достигает накопительного вознаграждения около 0,012 за 100 миллионов шагов.

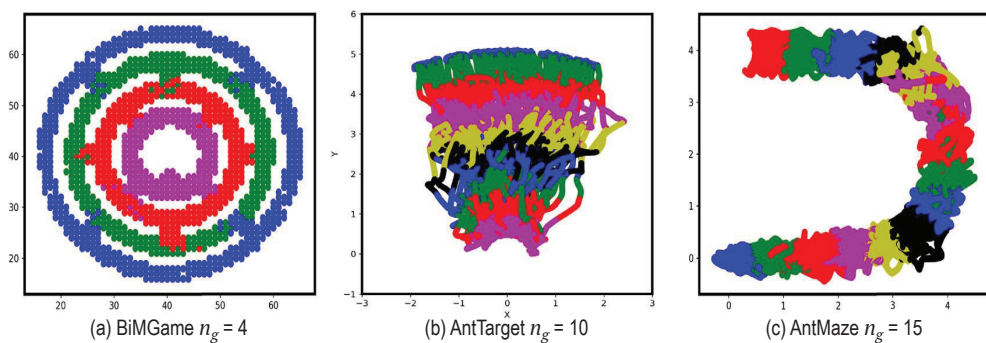


Рис. 3.15 ❖ (a) На этом рисунке представлены обозначенные разными цветами выученные подцели для трех задач. Обратите внимание, что для (b) и (c) нескольким подцелям назначается один и тот же цвет, но их можно различить по их пространственному расположению

Плохие результаты ValueReward и AggreVaTeD могут быть связаны с не-совершенной функцией ценности, обученной с помощью ограниченного количества примеров. В частности, при увеличении длины набора примеров вариации накопительного вознаграждения в начальном наборе состояний довольно велики. Это вносит значительную ошибку в функцию оценочного значения в начальных состояниях, что, в свою очередь, удерживает агента в ловушке некоторых локальных оптимумов, когда такие функции значений используются для управления процессом обучения.

Обсуждение. Из полученных результатов можно сделать следующие ключевые выводы: во-первых, метод обнаружения подцелей работает как для задач с присущими им узкими местами (например, BiMGame), так и без узких мест (например, AntTarget и AntMaze), но с временным порядком между группами состояний в экспертных примерах, что свойственно многим реальным сценариям. Во-вторых, дискретные вознаграждения с использованием подцелей работают намного лучше, чем непрерывные вознаграждения, основанные на функции ценности. Это может быть связано с тем, что функции ценности, извлеченные из ограниченного числа примеров, могут быть ошибочными, в то время как сегментация примеров на основе временного порядка может работать хорошо.

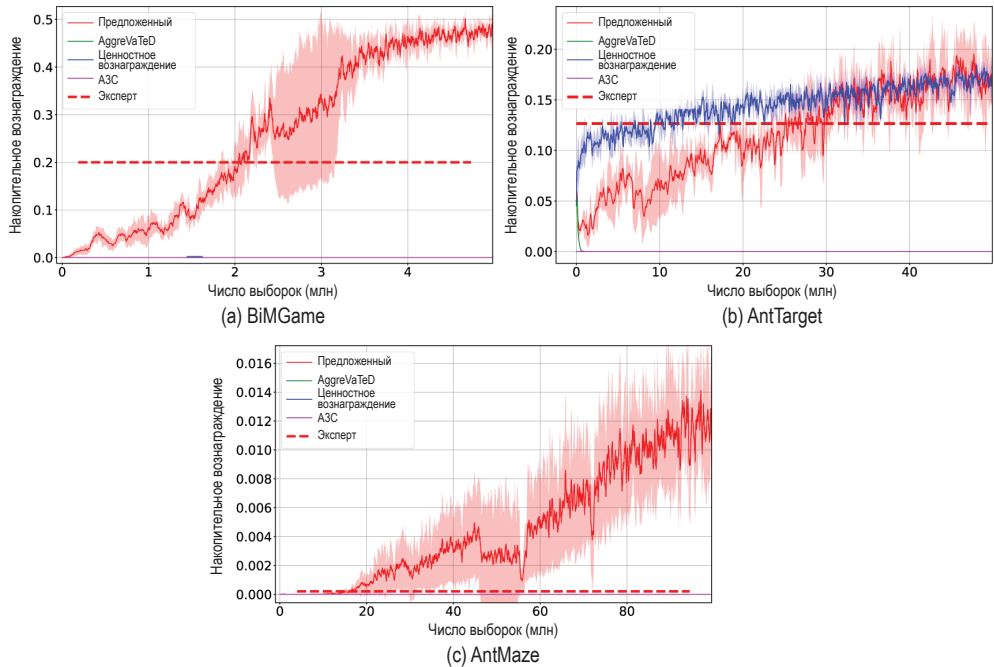


Рис. 3.16 ❖ На этом рисунке показано сравнение предложенного нами метода с аналогичными методами. Некоторые линии могут быть не видны, так как они перекрываются. В задачах (а) и (с) наш метод явно превосходит другие. В задаче (b), несмотря на то что ценностное вознаграждение работает лучше, наш метод в конечном итоге достигает того же уровня. Для справедливого сравнения мы не используем расширенные примеры вне набора для создания этих графиков. Поскольку некоторые алгоритмы вообще не обучаются, накапливая нулевое вознаграждение, такие линии пересекаются с осью x графиков и не видны

3.6. Выводы

Широко разрекламированные успехи машинного обучения во многом обусловлены наличием огромных объемов размеченных данных, используемых для обучения моделей. К сожалению, это абсолютно невозможно при обучении многих реальных моделей. Невозможно предусмотреть наперед, на этапе обучения, все сценарии, которые могут возникнуть во время работы поведенческой системы, а экспертные знания для расстановки меток могут просто отсутствовать. Например, если кошки и собаки могут быть правильно помечены на изображениях практически кем угодно, этого нельзя сказать про разметку различных видов птиц, типов растительности или медицинских диагнозов. Таким образом, обучение с ограниченным участием учителя или без него приобретает важное прикладное значение.

В этой главе был представлен обзор машинного обучения с ограниченным участием учителя (с ограниченной разметкой) для приложений в об-

ласти компьютерного зрения и робототехники. Обучение с учителем может принимать разные формы и зависит от предметной области. Поэтому мы рассмотрели несколько различных методов. Первый подход связан с активным обучением, где мы показали, как контекстную информацию, присутствующую в источниках данных, можно использовать для прогнозирования многих меток, которые в противном случае, возможно, придется добавлять вручную. Далее мы рассмотрели ситуации, когда подробная (сильная) разметка недоступна, но текстовые описания, сопровождающие видео, могут использоваться в качестве разновидности слабой разметки. Наша цель состоит в том, чтобы научиться локализовать соответствующий видеосегмент по этим описаниям. Обучающие и рабочие видео находятся в одном домене (предметной области). В следующем разделе мы рассмотрели проблему семантической сегментации изображения и то, как обученные модели из одного домена могут быть адаптированы к другому домену с минимальным участием учителя или даже без него. Наконец, разобрали обучение робота определенным задачам на ограниченных демонстрационных примерах, используя комбинацию имитации и обучения с подкреплением посредством выделения подцелей. Таким образом, мы исследовали различные парадигмы обучения и продемонстрировали их на примерах различных задач в компьютерном зрении и робототехнике.

Будущее машинного обучения будет основано на использовании ограниченных объемов обучающих данных. Исследователям и практикующим специалистам следует рассмотреть различные перспективные направления. Несколько примеров приведены ниже. Можем ли мы использовать для качественного обучения моделей огромный объем данных, загружаемых в социальные сети, при условии что эти данные содержат множество ошибок и пробелов? Можем ли мы перенести обученные модели из доменов, в которых доступна легкая разметка, в домены, требующие значительного опыта для разметки (например, разметка птиц и деревьев проста, но точная разметка видов птиц и деревьев требует серьезных знаний, которыми обладают лишь немногие люди)? Можем ли мы адаптировать обученные модели к неизвестным условиям, которые могут иметь решающее значение для безопасности (например, может ли система автономного вождения выявить опасную ситуацию на дороге, если не сталкивалась с ней при обучении)? Можно ли использовать машинное обучение в областях, где есть большие объемы данных, например в медицине, но обучение с учителем является очень дорогим (или даже невозможным)?

В этой главе были представлены лишь первые шаги в направлении решения вышеупомянутых проблем. Впереди нас ждет огромный объем работы. Хотя может показаться, что машинное обучение во многих случаях способно превзойти качество и надежность человеческих решений, оно в большинстве случаев нуждается в подготовленных обучающих данных. Присущая человеку способность рассуждать, абстрагироваться и определять общие принципы, которые можно перенести в другие домены, остается для машинного обучения далекой и труднодостижимой целью. На пути к этой цели необходимо решить много сложных проблем.

БЛАГОДАРНОСТИ

Работа была частично поддержана Национальным научным фондом США через грант 1724341, Управлением военно-морских исследований США через грант N00014-19-1-2264 и исследовательскими лабораториями Mitsubishi Electric.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Aggarwal Jake K., Ryoo Michael S.*, 2011. Human activity analysis: a review. *ACM Computing Surveys (CSUR)* 43 (3), 16.
- Andreas Jacob, Klein Dan, Levine Sergey*, 2017. Modular multitask reinforcement learning with policy sketches. In: *ICML*, pp. 166–175.
- Arandjelovic Relja, Gronat Petr, Torii Akihiko, Pajdla Tomas, Netvlad Josef Sivic*, 2016. Cnn architecture for weakly supervised place recognition. In: *CVPR*, pp. 5297–5307.
- Argall Brenna D., Chernova Sonia, Veloso Manuela, Browning Brett*, 2009. A survey of robot learning from demonstration. *RAS* 57 (5), 469–483.
- Bappy Jawadul H., Paul Sujoy, Roy-Chowdhury Amit K.*, 2016. Online adaptation for joint scene and object classification. In: *ECCV*. Springer, pp. 227–243.
- Bearman Amy, Russakovsky Olga, Ferrari Vittorio, Fei-Fei Li*, 2016. What's the point: semantic segmentation with point supervision. In: *ECCV*.
- Bilgic Mustafa, Getoor Lise*, 2009. Link-based active learning. In: *NIPS-Workshop*.
- Bojanowski Piotr, Lajugie Rémi, Grave Edouard, Bach Francis, Laptev Ivan, Ponce Jean, Schmid Cordelia*, 2015. Weakly-supervised alignment of video with text. In: *ICCV*. IEEE, pp. 4462–4470.
- Bojarski Mariusz, Del Testa Davide, Dworakowski Daniel, Firner Bernhard, Flepp Beat, Goyal Praseem, Jackel Lawrence D., Monfort Mathew, Muller Urs, Zhang Jiakai, et al.*, 2016. End to end learning for self-driving cars. *arXiv preprint. arXiv:1604.07316*.
- Carreira Joao, Zisserman Andrew*, 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In: *CVPR*. IEEE, pp. 4724–4733.
- Chakraborty Shayok, Balasubramanian Vineeth, Sun Qian, Panchanathan Sethuraman, Ye Jieping*, 2015. Active batch selection via convex relaxations with guaranteed solution bounds. *TPAMI* 37 (10), 1945–1958.
- Chang Chih-Chung, Lin Chih-Jen*, 2011. Libsvm: a library for support vector machines. *TIST* 2 (3), 27.
- Chang Wei-Lun, Wang Hui-Po, Peng Wen-Hsiao, Chiu Wei-Chen*, 2019. All about structure: adapting structural information across domains for boosting semantic segmentation. In: *CVPR*.
- Chen Liang-Chieh, Papandreou George, Kokkinos Iasonas, Murphy Kevin, Deeplab Alan L. Yuille*, 2016. Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*. *arXiv: 1606.00915 [abs]*.

- Chen Yuhua, Li Wen, Van Gool Road Luc*, 2018. Reality oriented adaptation for semantic segmentation of urban scenes. In: CVPR.
- Cheng Ching-An, Yan Xinyan, Wagener Nolan, Boots Byron*, 2018. Fast Policy Learning Through Imitation and Reinforcement. UAI.
- Choi Myung Jin, Lim Joseph J., Torralba Antonio, Willsky Alan S.*, 2010. Exploiting hierarchical context on a large database of object categories. In: CVPR. IEEE, pp. 129–136.
- Cordts Marius, Omran Mohamed, Ramos Sebastian, Rehfeld Timo, Enzweiler Markus, Benenson Rodrigo, Franke Uwe, Roth Stefan, Schiele Bernt*, 2016. The cityscapes dataset for semantic urban scene understanding. In: CVPR.
- Cristianini N., Ricci Elisa*, 2008. Support vector machines. Encyclopedia of algorithms.
- Cuong Nguyen Viet, Lee Wee Sun, Ye Nan, Chai Kian Ming A., Chieu Hai Leong*, 2013. Active learning for probabilistic hypotheses using the maximum Gibbs error criterion. In: NIPS, pp. 1457–1465.
- Du Liang, Tan Jingang, Yang Hongye, Feng Jianfeng, Xue Xiangyang, Zheng Qibao, Ye Xiaoqing, Zhang Xiaolin*, 2019. Ssf-dan: separated semantic feature based domain adaptation network for semantic segmentation. In: ICCV.
- Duan Yan, Chen Xi, Houthoof Rein, Schulman John, Abbeel Pieter*, 2016. Benchmarking deep reinforcement learning for continuous control. In: ICML, pp. 1329–1338.
- Dubey Rachit, Agrawal Pulkit, Pathak Deepak, Griffiths Thomas L., Efros Alexei A.*, 2018. Investigating human priors for playing video games. In: ICML.
- Fujishige Satoru, Hayashi Takumi, Isotani Shiguo*, 2006. The minimum-norm-point algorithm applied to submodular function minimization and linear programming. In: Citeseer.
- Galleguillos Carolina, Rabinovich Andrew, Belongie Serge*, 2008. Object categorization using co-occurrence, location and appearance. In: CVPR. IEEE, pp. 1–8.
- Goodfellow Ian J., Pouget-Abadie Mehdi, Mirza Jean, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, Bengio Yoshua*, 2014. Generative adversarial nets. In: NIPS.
- Hartmann Glenn, Grundmann Matthias, Hoffman Judy, Tsai David, Kwatra Vivek, Madani Omid, Vijayanarasimhan Sudheendra, Essa Irfan, Rehg James, Sukthankar Rahul*, 2012. Weakly supervised learning of object segmentations from web-scale video. In: ECCVW. Springer, pp. 198–208.
- Hasan Mahmudul Paul, Sujoy Mourikis Anastasios I., Roy-Chowdhury Amit K.*, 2018. Context-aware query selection for active learning in event recognition. T-PAMI.
- Hasan Mahmudul, Roy-Chowdhury Amit K.*, 2015. Context aware active learning of activity recognition models. In: ICCV. IEEE, pp. 4543–4551.
- He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian*, 2016. Deep residual learning for image recognition. In: CVPR.
- Heilbron Fabian Caba, Escorcia Victor, Ghanem Bernard, Niebles Juan Carlos*, 2015. Activitynet: a large-scale video benchmark for human activity understanding. In: CVPR. IEEE, pp. 961–970.
- Held David, Geng Xinyang, Florensa Carlos, Abbeel Pieter*, 2017. Automatic goal generation for reinforcement learning agents. In: ICML.

- Hoffman Judy, Tzeng Eric, Park Taesung, Zhu Jun-Yan, Isola Phillip, Saenko Kate, Efros Alexei A., Cycada Trevor Darrell, 2018. Cycle-consistent adversarial domain adaptation. In: ICML.
- Hoffman Judy, Wang Dequan, Yu Fisher, Darrell Trevor, 2016. Fcns in the wild: pixel-level adversarial and constraint-based adaptation. CoRR. arXiv:1612.02649 [abs].
- Holub Alex, Perona Pietro, Burl Michael C., 2008. Entropy-based active learning for object recognition. In: CVPRWorkshops. IEEE, pp. 1–8.
- Hu Xia, Tang Jiliang, Gao Huiji, Liu Actnet Huan, 2013. Active learning for networked texts in microblogging. In: SDM. SIAM, pp. 306–314.
- Huang De-An, Fei-Fei Li, Nieves Juan Carlos, 2016. Connectionist temporal modeling for weakly supervised action labeling. In: ECCV. Springer, pp. 137–153.
- Hung Wei-Chih, Tsai Yi-Hsuan, Liou Yan-Ting, Lin Yen-Yu, Yang Ming-Hsuan, 2018. Adversarial learning for semi-supervised semantic segmentation. In: BMVC.
- Idrees Haroon, Zamir Amir R., Jiang Yu-Gang, Gorban Alex, Laptev Ivan, Sukthankar Rahul, Shah Mubarak, 2017. The thumos challenge on action recognition for videos «in the wild». CVIU 155, 1–23.
- Jain Ashesh, Zamir Amir R., Savarese Silvio, Saxena Ashutosh, 2015. Structural-rnn: deep learning on spatiotemporal graphs. In: CVPR.
- Jie Zequn, Wei Yunchao, Jin Xiaojie, Feng Jiashi, Liu Wei, 2017. Deep self-taught learning for weakly supervised object localization. In: CVPR.
- Käding Christoph, Freytag Alexander, Rodner Erik, Perino Andrea, Denzler Joachim, 2016. Large-scale active learning with approximations of expected model output changes. In: GCPR. Springer, pp. 179–191.
- Kanazawa Angjoo, Jacobs David W., Chandraker Manmohan, 2016. WarpNet: weakly supervised matching for single-view reconstruction. In: CVPR, pp. 3253–3261.
- Khoreva Anna, Benenson Rodrigo, Hosang Jan, Hein Matthias, Schiele Bernt, 2017. Simple does it: weakly supervised instance and semantic segmentation. In: CVPR.
- Khoreva Anna, Benenson Rodrigo, Omran Mohamed, Hein Matthias, Schiele Bernt, 2016. Weakly supervised object boundaries. In: CVPR, pp. 183–192.
- Koller Daphne, Friedman Nir, 2009. Probabilistic Graphical Models: Principles and Techniques. MIT Press.
- Koppula Hema Swetha, Gupta Rudhir, Saxena Ashutosh, 2013. Learning human activities and object affordances from rgb-d videos. IJRR 32 (8), 951–970.
- Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E., 2012. Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105.
- Lapedriza Agata, Pirsavash Hamed, Bylinskii Zoya, Torralba Antonio, 2013. Are all training examples equally valuable? arXiv preprint. arXiv:1311.6510.
- Levine Sergey, Finn Chelsea, Darrell Trevor, Abbeel Pieter, 2016. End-to-end training of deep visuomotor policies. JMLR 17 (1), 1334–1373.
- Li Dong, Huang Jia-Bin, Li Yali, Wang Shengjin, Yang Ming-Hsuan, 2016. Weakly supervised object localization with progressive domain adaptation. In: CVPR, pp. 3512–3520.
- Li Xianglin, Guo Runqiu, Cheng Jun, 2012. Incorporating Incremental and Active Learning for Scene Classification. In: ICMLA, vol. 1. IEEE, pp. 256–261.
- Li Xin, Guo Yuhong, 2013. Adaptive active learning for image classification. In: CVPR, pp. 859–866.

- Li Xin, Guo Yuhong*, 2014. Multi-level adaptive active learning for scene classification. In: ECCV. Springer, pp. 234–249.
- Li Yunsheng, Yuan Lu, Vasconcelos Nuno*, 2019. Bidirectional learning for domain adaptation of semantic segmentation. In: CVPR.
- Lian Qing, Lv Fengmao, Duan Lixin, Gong Boqing*, 2019. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: a non-adversarial approach. In: ICCV.
- Mac Aodha Oisin, Campbell Neill, Kautz Jan, Brostow Gabriel*, 2014. Hierarchical subquery evaluation for active learning on a graph. In: CVPR. IEEE, pp. 564–571.
- McCormick S. Thomas*, 2005. Submodular function minimization. Handbooks in Operations Research and Management Science 12, 321–391.
- Mnih Volodymyr, Badia Mehdi Mirza, Adria Puigdomenech, Graves Alex, Lillicrap Timothy, Harley Tim, Silver David, Kavukcuoglu Koray*, 2016. Asynchronous methods for deep reinforcement learning. In: ICML, pp. 1928–1937.
- Mnih Volodymyr, Kavukcuoglu Koray, Silver David, Rusu Andrei A., Veness Joel, Bellemare Marc G., Graves Alex, Riedmiller, Martin, Fidjeland Andreas K., Ostrovski Georg, et al.*, 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529.
- Murez Zak, Kolouri Soheil, Kriegman David, Ramamoorthi Ravi, Kim Kyungnam*, 2018. Image to image translation for domain adaptation. In: CVPR.
- Ng Andrew Y., Harada Daishi, Russell Stuart*, 1999. Policy invariance under reward transformations: theory and application to reward shaping. In: ICML, vol. 99, pp. 278–287.
- Nguyen Phuc, Liu Ting, Prasad Gautam, Han Bohyung*, 2018. Weakly supervised action localization by sparse temporal pooling network. In: CVPR.
- Oh Sangmin, Hoogs Anthony, Perera Amitha, Cuntoor Naresh, Chen Chia-Chih, Taek Lee Jong, Mukherjee Saurajit, Aggarwal J. K., Lee Hyungtae, Davis Larry, et al.*, 2011. A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR. IEEE, pp. 3153–3160.
- Panda Rameswar, Das Abir, Wu Ziyang, Ernst Jan, Roy-Chowdhury Amit K.*, 2017. Weakly supervised summarization of web videos. In: ICCV, pp. 3657–3666.
- Paul Sujoy, Bappy Jawadul H., Roy-Chowdhury Amit K.*, 2016. Efficient selection of informative and diverse training samples with applications in scene classification. In: ICIP. IEEE, pp. 494–498.
- Paul Sujoy, Bappy Jawadul H., Roy-Chowdhury Amit K.*, 2017. Non-uniform subset selection for active learning in structured data. In: CVPR, pp. 6846–6855.
- Paul Sujoy, Roy Sourya, Roy-Chowdhury Amit K.*, 2018. W-talc: weakly-supervised temporal activity localization and classification. In: ECCV, pp. 563–579.
- Paul Sujoy, Tsai Yi-Hsuan, Schuster Samuel, Roy-Chowdhury Amit K., Chandraker, Manmohan* 2020. Domain adaptive semantic segmentation using weak labels. In: ECCV.
- Paul Sujoy, Vanbaars Jeroen, Roy-Chowdhury Amit*, 2019. Learning from trajectories via subgoal discovery. In: NeurIPS, pp. 8411–8421.
- Richter Stephan R., Vineet Vibhav, Roth Stefan, Koltun Vladlen*, 2016. Playing for data: ground truth from computer games. In: ECCV.
- Ross Stéphane, Gordon Geoffrey, Bagnell Drew*, 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In: AISTATS, pp. 627–635.

- Sadat Saleh Fatemeh, Aliakbarian Mohammad Sadegh, Salzmann Mathieu, Petersson Lars, Alvarez Jose M.*, 2018. Effective use of synthetic data for urban scene semantic segmentation. In: ECCV.
- Ugm Mark Schmidt*, 2007. A Matlab toolbox for probabilistic undirected graphical models.
- Schulman John, Moritz Philipp, Levine Sergey, Jordan Michael, Abbeel Pieter*, 2015. High-dimensional continuous control using generalized advantage estimation. In: ICLR.
- Sen Prithviraj, Getoor Lise*, 2003. Link-based classification. In: ICML.
- Sen Prithviraj, Namata Galileo, Bilgic Mustafa, Getoor Lise, Galligher Brian, Eliassi-Rad Tina*, 2008. Collective classification in network data. *AI Magazine* 29 (3), 93.
- Settles Burr*, 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6 (1), 1–114.
- Settles Burr, Craven Mark*, 2008. An analysis of active learning strategies for sequence labeling tasks. In: EMNLP. Association for Computational Linguistics, pp. 1070–1079.
- Shen Zhiqiang, Li Jianguo, Su Zhou, Li Minjun, Chen Yurong, Jiang Yu-Gang, Xue Xiangyang*, 2017. Weakly supervised dense video captioning. In: CVPR, vol. 2, p. 10.
- Silver David, Huang Aja, Maddison Chris J., Guez Arthur, Sifre Laurent, VanDen Driessche George, Schrittwieser Julian, Antonoglou Ioannis, Panneershelvam Veda, Lanctot Marc, et al.*, 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529 (7587), 484.
- Singh Avi, Yang Larry, Gplac Sergey Levine*, 2017. Generalizing vision-based robotic skills using weakly labeled images. In: ICCV.
- Sun Wen, Bagnell J. Andrew, Boots Byron*, 2018. Truncated horizon policy search: combining reinforcement learning & imitation learning. In: ICLR.
- Sun Wen, Venkatraman Arun, Gordon Geoffrey J., Boots Byron, Bagnell J. Andrew*, 2017. Deeply aggravated: differentiable imitation learning for sequential prediction. In: ICML, pp. 3309–3318.
- Todorov Emanuel*, 2014. Convex and analytically-invertible dynamics with contacts and constraints: theory and implementation in mujoco. In: ICRA, pp. 6054–6061.
- Tran Du, Bourdev Lubomir, Fergus Rob, Torresani Lorenzo, Paluri Manohar*, 2015. Learning spatiotemporal features with 3d convolutional networks. In: ICCV. IEEE, pp. 4489–4497.
- Tsai Yi-Hsuan, Hung Wei-Chih, Schulter Samuel, Sohn Kihyuk, Yang Ming-Hsuan, Chandraker Manmohan*, 2018. Learning to adapt structured output space for semantic segmentation. In: CVPR.
- Tsai Yi-Hsuan, Sohn Kihyuk, Schulter Samuel, Chandraker Manmohan*, 2019. Domain adaptation for structured output via discriminative patch representations. In: ICCV.
- Tulyakov Stepan, Ivanov Anton, Fleuret Francois*, 2017. Weakly supervised learning of deep metrics for stereo reconstruction. In: CVPR, pp. 1339–1348.
- van Baar Jeroen, Sullivan Alan, Cordorel Radu, Jha Devesh, Romeres Diego, Nikovski Daniel*, 2018. Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics. In: ICRA.

- Wang Botao, Lin Dahua, Xiong Hongkai, Zheng Y. F., 2016a. Joint inference of objects and scenes with efficient learning of text-object-scene relations. *TMM* 18 (3), 507–520.
- Wang Limin, Xiong Yuanjun, Wang Zhe, Qiao Yu, Lin Dahua, Tang Xiaoou, Van Gool Luc, 2016b. Temporal segment networks: towards good practices for deep action recognition. In: *ECCV*. Springer, pp. 20–36.
- Wang Limin, Xiong Yuanjun, Lin Dahua, Van Gool Luc, 2017. Untrimmednets for weakly supervised action recognition and detection. In: *CVPR*.
- Wang Zhenhua, Shi Qinfeng, Shen Chunhua, Van Den Hengel Anton, 2013. Bilinear programming for human activity recognition with unknown mrf graphs. In: *CVPR*, pp. 1690–1697.
- Wu Zuxuan, Han Xintong, Lin Mustafa Gkhan Uzunbas Yen-Liang, Goldstein Tom, Nam Lim Ser, Dcan Larry S. Davis, 2018. Dual channel-wise alignment networks for unsupervised scene adaptation. In: *ECCV*.
- Xiao Jianxiong, Hays James, Ehinger Krista A., Oliva Aude, Torralba Antonio, 2010. Sun database: large-scale scene recognition from abbey to zoo. In: *CVPR*. IEEE, pp. 3485–3492.
- Xu Huijuan, Das Abir, Saenko Kate, 2017. R-c3d: region convolutional 3d network for temporal activity detection. In: *ICCV*, vol. 6, p. 8.
- Yan Yan, Xu Chenliang, Cai Dawen, Corso Jason, 2017. Weakly supervised actor-action segmentation via robust multi-task ranking. *CVPR* 48, 61.
- Yao Bangpeng, Fei-Fei Li, 2010. Modeling mutual context of object and human pose in human-object interaction activities. In: *CVPR*. IEEE, pp. 17–24.
- Yao Jian, Fidler Sanja, Urtasun Raquel, 2012. Describing the scene as a whole: joint object detection, scene classification and semantic segmentation. In: *CVPR*. IEEE, pp. 702–709.
- Zhang Chicheng, Chaudhuri Kamalika, 2015. Active learning from weak and strong labelers. In: *NIPS*, pp. 703–711.
- Zhang Yang, David Philip, Gon Boqing, 2017. Curriculum domain adaptation for semantic segmentation of urban scenes. In: *ICCV*.
- Zhao Hengshuang, Shi Jianping, Qi Xiaojuan, Wang Xiaogang, Jia Jiaya, 2017a. Pyramid scene parsing network. In: *CVPR*.
- Zhao Yue, Xiong Yuanjun, Wang Limin, Wu Zhirong, Tang Xiaoou, Lin Dahua, 2017b. Temporal action detection with structured segment networks. In: *ICCV*.
- Zhong Bineng, Yao Hongxun, Chen Sheng, Ji Rongrong, Chin Tat-Jun, Wang Hanzhi, 2014. Visual tracking via weakly supervised learning from multiple imperfect oracles. *Pattern Recognition* 47 (3), 1395–1410.
- Zhou Bolei, Lapedriza Agata, Xiao Jianxiong, Torralba Antonio, Oliva Aude, 2014. Learning deep features for scene recognition using places database. In: *NIPS*, pp. 487–495.
- Zhou Zhi-Hua, 2004. Multi-Instance Learning: A Survey. Tech. Rep. Department of Computer Science & Technology, Nanjing University.
- Zou Yang, Yu Zhiding, Vijaya Kumar B. V. K., Wang Jinsong, 2018. Domain adaptation for semantic segmentation via class-balanced self-training. In: *ECCV*.

ОБ АВТОРАХ ГЛАВЫ

Суджой Пол в настоящее время работает в Google Research. Он получил докторскую степень в области электроники и вычислительной техники в Калифорнийском университете в Риверсайде и степень бакалавра в области электроники и телекоммуникаций в Университете Джадавпур. Область его научных интересов включает в себя компьютерное зрение и машинное обучение с упором на семантическую сегментацию, распознавание действий человека, адаптацию предметной области, обучение со слабой разметкой, активное обучение, обучение с подкреплением.

Амит Рой-Чоудхури – профессор и научный сотрудник факультета электроники и вычислительной техники Bourns Family, директор Центра робототехники и интеллектуальных систем и факультета информатики и вычислительной техники Калифорнийского университета в Риверсайде (UCR). Он возглавляет группу видеовычислений, работающую над основополагающими принципами компьютерного зрения, обработки изображений и статистического обучения с применением в киберфизических, автономных и интеллектуальных системах. Он опубликовал более 200 работ в рецензируемых журналах и на конференциях. Является членом IEEE и IAPR, получил награду за консультирование и наставничество в области докторских диссертаций в 2019 г. от UCR и награду ECE Distinguished Alumni Award от Мэрилендского университета в Колледж-Парке.

Глава 4

Эффективные методы глубокого обучения

Авторы главы:

Хан Цай, Цзи Линь и Сун Хань,

Массачусетский технологический институт,

Кембридж, Массачусетс, США.

Все авторы внесли равный вклад в эту работу
и перечислены в алфавитном порядке.

Краткое содержание главы:

- различные методы сжатия моделей, такие как прореживание, факторизация, квантование и разработка эффективных моделей;
- снижение стоимости проектирования путем подбора нейронной архитектуры, автоматического прореживания и дискретизации, которые могут превзойти ручное проектирование и требуют минимальных усилий со стороны человека;
- метод эффективной работы со многими аппаратными платформами и ограничениями эффективности без повторения дорогостоящих этапов поиска и переобучения.

4.1. СЖАТИЕ МОДЕЛИ

Сжатие глубокой нейронной сети – это действенный способ повысить эффективность логического вывода. Методы сжатия включают *прореживание*¹ *параметров* (parameter pruning) для удаления избыточных весов, *низкоранговую факторизацию* (low-rank factorization) для уменьшения сложности, *квантование весов* (weight quantization) для уменьшения точности весов и размера модели, а также *дистилляцию знаний* (knowledge distillation) для переноса *скрытых знаний* (dark knowledge) из больших моделей в меньшие. В заключение мы обсудим методы автоматического поиска хорошей политики сжатия без участия человека.

¹ В публикациях по машинному обучению часто встречается калька «прунинг», но в этой книге мы будем использовать перевод. – Прим. перев.

4.1.1. Прореживание параметров

Глубокие нейронные сети обычно чрезмерно параметризованы, т. е. обладают излишним числом параметров. Прореживание удаляет избыточные элементы нейронных сетей (рис. 4.1), чтобы уменьшить размер модели и объем вычислений.

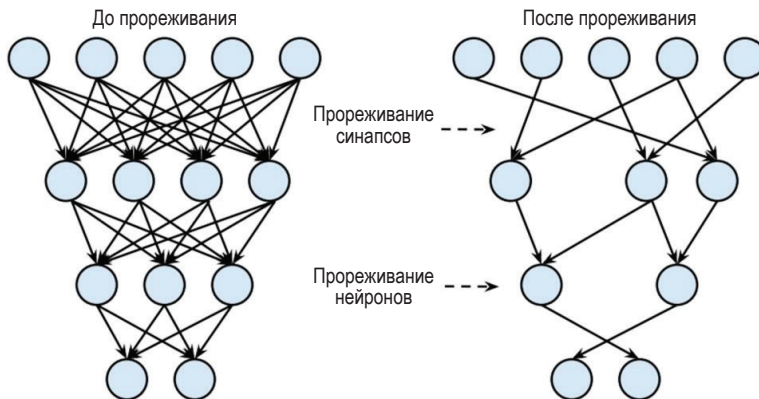


Рис. 4.1 ❖ Синапсы и нейроны до и после прореживания (Han et al., 2015b).

Обозначения

Мы рассматриваем сверточные слои в глубоких нейронных сетях, которые являются наиболее вычислительно затратными компонентами. Веса одного сверточного слоя составляют 4-мерный тензор $n \times c \times k_h \times k_w$, где n – количество фильтров (т. е. выходных каналов), c – количество каналов (т. е. входных каналов) и k_h, k_w – размер ядра (обычно симметричный, т. е. $k_h = k_w$). Веса одного слоя можно рассматривать как несколько *фильтров* (трехмерные тензоры $c \times k_h \times k_w$), каждый из которых соответствует выходному каналу; или рассматривать как несколько *каналов* (трехмерные тензоры $n \times k_h \times k_w$), каждый из которых соответствует входному каналу. Каждый тензор $k_h \times k_w$ является ядром; в сверточном слое имеется $n \times c$ ядер.

Гранулярность

Прореживание можно выполнять с разной степенью *гранулярности* (детализации) (Mao et al., 2017) (рис. 4.2).

Мелкомодульное прореживание (fine-grained pruning) удаляет отдельные элементы из тензора весов. Первыми методами были Optimal Brain Damage (LeCun et al., 1989) и Optimal Brain Surgeon (Hassibi and Stork, 1993), которые уменьшали количество связей на основе гессииана функции потерь. Хан и др. (Han et al., 2015) предложили трехэтапный метод «обучение–обрезка–перевоспитывание» для удаления избыточных связей в глубокой нейронной сети. Этот метод уменьшил количество параметров AlexNet в 9 раз, а VGG-16 в 13 раз без потери точности. Шринивас и Бабу (Srinivas, Babu, 2015) предложили

метод прореживания без данных для удаления избыточных нейронов. При мелкомодульном прореживании набор весов для прореживания может быть выбран произвольно, с его помощью можно достичь очень высокого коэффициента сжатия CNN (Han et al., 2015), RNN (Giles, Omlin, 1994), LSTM (Han et al., 2017), трансформеров (Cheong, Daniel, 2019) и т. д. без ущерба для точности.

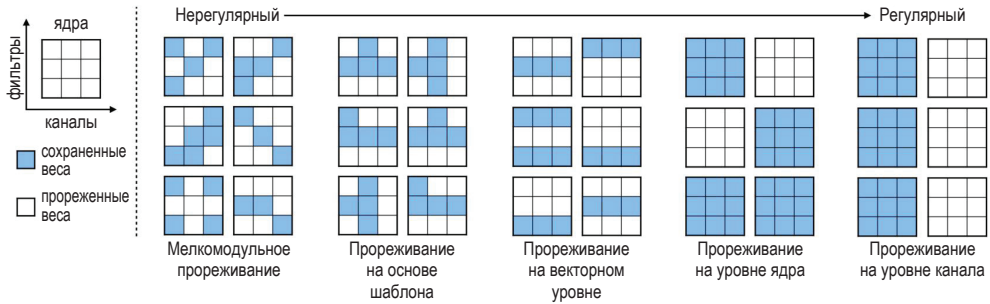


Рис. 4.2 ❖ Различные степени гранулярности при прореживании весов (на основе рисунка из Mao et al., 2017)

Прореживание на основе шаблонов (pattern-based pruning) – это особый вид мелкомодульного прореживания, который обеспечивает лучшее аппаратное ускорение с оптимизацией компилятора (Ma et al., 2020; Tan et al., 2020b; Niu et al., 2020). Если взять в качестве примера свертки 3×3 , прореживание на основе шаблона назначает фиксированный набор масок каждому из ядер 3×3 . Количество масок обычно ограничено (4–6) для обеспечения аппаратной эффективности. Каждый шаблон маски имеет фиксированное количество удаленных элементов для каждого ядра (пять удаленных элементов из девяти на рис. 4.2). Шаблоны определяются эвристикой (Ma et al., 2020) или кластеризацией на основе предварительно обученных весов (Niu et al., 2020). Несмотря на мелкомодульный шаблон прореживания внутри ядра, прореживание на основе шаблонов можно ускорить с помощью оптимизации компилятора путем переупорядочения циклов вычислений, что снижает накладные расходы на логику управления.

Крупномодульное прореживание (coarse-grained pruning), или *структурированное прореживание* (structured pruning), удаляет регулярный тензорный блок для повышения эффективности оборудования. В зависимости от размера блока могут быть удалены целые векторы (Mao et al., 2017), ядра (Mao et al., 2017; Niu et al., 2020) или каналы (He et al., 2017; Wen et al., 2016; Li et al., 2016; Molchanov et al., 2016) (рис. 4.2). Грубое крупномодульное прореживание, такое как прореживание каналов, может обеспечить прямое аппаратное ускорение на графических процессорах с использованием стандартных библиотек глубокого обучения, но обычно приводит к заметному падению точности по сравнению с мелкомодульным прореживанием (Li et al., 2016). Прореживание с меньшей модульностью обычно приводит к меньшему падению точности при той же степени сжатия (Mao et al., 2017).

Аппаратное ускорение

Вообще говоря, более регулярные схемы прореживания удобнее для оборудования, что упрощает ускорение логического вывода на существующем оборудовании, таком как графические процессоры; в то время как нерегулярные схемы лучше сохраняют точность при той же степени сжатия. С помощью специализированных аппаратных ускорителей (Han et al., 2016; Chen et al., 2016; Han et al., 2017; Chen et al., 2019a; Zhang et al., 2016a; Yu et al., 2017) и методов оптимизации, ориентированных на компиляторы (Ma et al., 2020; Niu et al., 2020), также можно получить значительное ускорение для нерегулярных методов прореживания.

Критерий значимости

После выбора уровня модульности прореживания следующим фактором влияния на производительность сжатой модели является определение весовых коэффициентов, которые должны быть сокращены. Было предложено несколько эвристических *критериев значимости* (importance criteria) для оценки важности каждого веса *после* обучения модели; менее важные веса обрезаются в соответствии с критериями. Самый простой эвристический подход основан на величинах, то есть абсолютных значениях весов (Han et al., 2015):

$$\text{Значимость} = |w|,$$

где веса большей величины считаются более значимыми. Это также распространяется на крупномодульное прореживание, такое как прореживание каналов, где в качестве критерия используется тензорная норма:

$$\text{Значимость} = \|\mathbf{W}\|_2.$$

К другим критериям относятся производные второго порядка (т. е. гессиан функции потерь) (LeCun et al., 1989; Hassibi, Stork, 1993), разложение Тейлора (Molchanov et al., 2017), выходная чувствительность (Engelbrecht, 2001) и др.

Недавно Франкл и Карбин (Frankle, Carbin, 2018) предложили *гипотезу лотерейного билета* для нахождения в плотных, случайным образом инициализированных глубоких сетях разреженных подсетей, которые можно обучить для достижения той же точности. Эксперименты показывают, что метод может находить разреженные подсети с весами менее 10–20 %, достигая того же уровня точности в MNIST (LeCun et al., 2010) и CIFAR (Крижевский и Хинтон, 2009). Позже метод был масштабирован до более масштабных применений (например, ResNet-50 и Inception-v3 в ImageNet), где разреженную подсеть можно найти на ранней стадии обучения (Frankle et al., 2020), а не при инициализации.

Методы обучения

Прямое удаление весов в глубоких нейронных сетях значительно ухудшит точность при большой степени сжатия. Следовательно, для восстановления потерянного качества требуется некоторое дообучение (тонкая настройка). После прореживания можно выполнить тонкую настройку, чтобы восстано-

вить падение производительности (He et al., 2017). Стратегию можно расширить до *итеративного прореживания* (Han et al., 2015) (рис. 4.3), когда для дальнейшего повышения точности выполняется несколько итераций прореживания и тонкой настройки. Чтобы избежать ошибочного удаления весов, динамическое прореживание (Guo et al., 2016) включает рассечение связей в рабочий процесс и обеспечивает постоянное обслуживание сети. *Прореживание на ходу* (runtime pruning, Lin et al., 2017) выбирает коэффициент сжатия сети в соответствии с каждой входной выборкой, назначая более агрессивную стратегию сжатия для более простых выборок, что еще больше улучшает компромисс между точностью вычислений и размером сети.

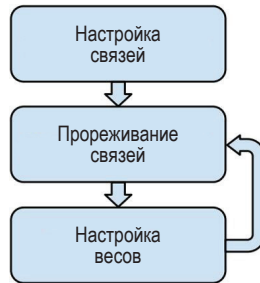


Рис. 4.3 ❖ Трехэтапный конвейер обучения с итеративной обрезкой (Han et al., 2015b)

В другой реализации обучают компактные DNN, используя *ограничения разреженности* (sparsity constraint). Ограничения разреженности обычно реализуются с помощью регуляризации L_0 -, L_1 - или L_2 -норм, применяемой к весам, которые добавляются к потерям при обучении для совместной оптимизации. Хан и др. (Han et al., 2015) применили регуляризацию L_1/L_2 к каждому отдельному весу во время обучения. Лебедев и Лемпицкий (Lebedev, Lempitsky, 2016) применили групповые ограничения разреженности к сверточным фильтрам для достижения структурированной разреженности.

4.1.2. Низкоранговая факторизация

Низкоранговая факторизация (low-rank factorization) использует декомпозицию матрицы/тензора, чтобы уменьшить сложность сверточных или полносвязных слоев в глубоких нейронных сетях. Идея использования фильтров низкого ранга для ускорения свертки уже давно исследуется в области обработки сигналов.

Наиболее широко используется декомпозиция Truncated SVD (Golub, Van Loan, 1996), которая эффективно ускоряет работу полносвязных слоев (Xue et al., 2013; Denton et al., 2014; Girshick, 2015). Для полносвязного слоя с весом $W \in \mathbb{R}^{m \times k}$ SVD определяется как $W = U S V^T$, где $U \in \mathbb{R}^{m \times m}$, $S \in \mathbb{R}^{m \times k}$, $V \in \mathbb{R}^{k \times k}$. Здесь S – диагональная матрица с сингулярными значениями на диагонали.

Если вес попадает в структуру низкого ранга, его можно аппроксимировать, сохранив только t самых больших элементов S , где $t \ll \min(m, k)$. Вычисление Wx может быть уменьшено с $O(mk)$ до $O(mt + tk)$ для каждой выборки.

Для сверточных весов 4D предложено (Jaderberg et al., 2014) разложить ядра $k \times k$ на ядра $1 \times k$ и $k \times 1$; этот подход также был принят в проекте Inception-V3 (Szegedy, Vanhoucke, 2016). Чжан и др. (Zhang et al., 2016) предложили разложить сверточный вес $n \times c \times k \times k$ на $n' \times c \times k \times k$ и $n \times n' \times 1 \times 1$, где $n' \ll n$. Для декомпозиции ядер более высокой размерности, таких как сверточные веса, может применяться каноническая полиадическая декомпозиция (Lebedev et al., 2014). Этот подход выполняет низкоранговую CP-декомпозицию четырехмерного тензора ядра свертки в сумму небольшого числа тензоров первого ранга. Во время вывода исходная свертка заменяется последовательностью из четырех сверточных слоев с меньшими ядрами. Ким и др. (Kim et al., 2015) использовали декомпозицию Такера (известную как расширение SVD более высокого порядка) для факторизации сверточных ядер, получая более высокую степень сжатия по сравнению с применением SVD.

4.1.3. Квантование

Квантование сети (network quantization) сжимает сеть, уменьшая количество битов на вес, необходимое для представления глубокой сети. После квантования сеть также может демонстрировать более высокую скорость вывода с аппаратной поддержкой.

Схемы округления

Чтобы преобразовать вес, представленный с полной точностью (32-битное значение с плавающей запятой), в значение с более низкой точностью, выполняется округление, отображающее значение с плавающей запятой в один из сегментов квантования.

В ранних работах (Han et al., 2015; Gong et al., 2014; Wu et al., 2016) применялась *кластеризация k -средних* для нахождения общих весов для каждого слоя настроенной¹ (trained) сети; все веса, попадающие в один и тот же кластер, будут иметь одинаковый вес. В частности, при разбиении n исходных весов $W = \{w_1, w_2, \dots, w_n\}$ на k кластеров $C = \{c_1, c_2, \dots, c_k\}$, $n \gg k$, мы минимизируем *внутрикластерную сумму квадратов* (within-cluster sum of squares, WCSS):

$$\operatorname{argmin}_C \sum_{i=1}^k \sum_{w \in c_i} |w - c_i|^2. \quad (4.1)$$

¹ В случае когда мы говорим о сжатии моделей с целью оптимизации, между терминами learning (обучение) и training (настройка) появляется тонкое, но важное различие. В контексте оптимизации под *настройкой* часто понимают обучение модели на специально подобранных образцах, что облегчает и улучшает последующее сжатие модели без потери качества. Иногда под *настройкой* понимают начальное обучение модели на базовом домене с последующим дообучением, а иногда в зарубежной литературе learning и training являются просто синонимами. – Прим. перев.

Квантование на основе кластеризации k -средних можно комбинировать с прореживанием и кодированием Хаффмана для выполнения сжатия модели (Han et al., 2015) (рис. 4.4). Этот метод позволяет уменьшить размер модели VGG-16 в 49 раз без потери точности.

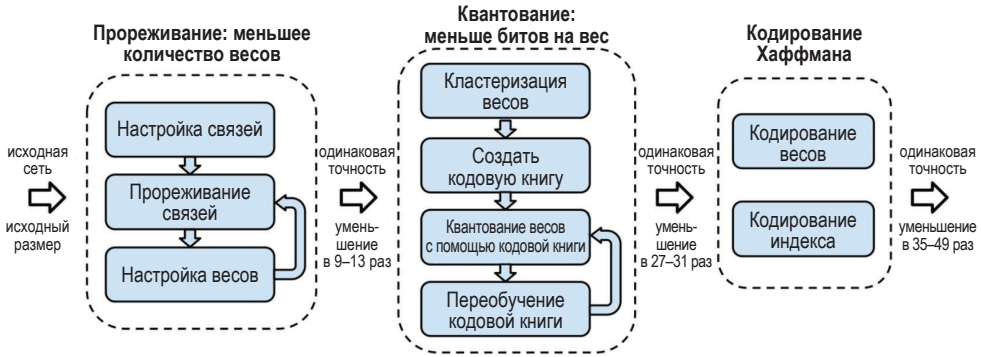


Рис. 4.4 ❖ Трехэтапный процесс сжатия: прореживание, квантование и кодирование Хаффмана (Han et al., 2015)

Линейное/равномерное квантование (Vanhoucke et al., 2011; Jacob et al., 2017) непосредственно округляет значение с плавающей запятой до ближайших квантованных значений после усечения диапазона; градиент распространяется с использованием приближения STE (Bengio et al., 2013). Предположим, что диапазон отсечения равен $[a, b]$, а количество уровней квантования равно n , тогда прямой процесс квантования значения с плавающей запятой x в квантованное значение q выглядит так:

$$\text{clamp}(r, a, b) = \min(\max(x, a), b); \quad (4.2)$$

$$s(a, b, n) = \frac{b - a}{n - 1}; \quad (4.3)$$

$$q = \text{round}\left(\frac{\text{clamp}(r, a, b) - a}{s(a, b, n)}\right)s(a, b, n) + a. \quad (4.4)$$

Градиент обратного распространения вычисляется при помощи

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial \mathcal{L}}{\partial x}. \quad (4.5)$$

Помимо применения значения усечения a, b , в некоторых работах (Zhou et al., 2016) используются функции активации, такие как \tanh , для переноса диапазона весов в диапазон $[-1, 1]$, что упрощает квантование.

Разрядность

Мы можем найти компромисс между размером и точностью модели, используя разную разрядность чисел (битовую точность модели). Более низкая

разрядность может привести к уменьшению размера модели, но за это, возможно, придется заплатить снижением ее качества. Сети с полной точностью используют 32-разрядные числа с плавающей запятой как для весов, так и для активаций. Сети половинной точности используют 16-разрядные числа с плавающей запятой для уменьшения размера модели вдвое. Квантование до 8-разрядных целых чисел как для весов, так и для активаций (Jacob et al., 2017) широко используется в целочисленных арифметических операциях, которые можно ускорить на графических процессорах, обычных процессорах, смартфонах и т. д.

К категории низкой точности относятся *сети с троичными весами* (Li et al., 2016), где веса квантуются до значений $\{-1, 0, 1\}$ или $\{-E, 0, E\}$ (здесь E – среднее значение абсолютного веса). Настраиваемое троичное квантование (Zhu et al., 2016) использует два обучаемых коэффициента масштабирования полной точности W_l^p , W_l^n для каждого слоя l и квантует веса до $\{-W_l^n, 0, +W_l^p\}$. Этот метод позволяет квантовать AlexNet на ImageNet без потери точности.

Крайним случаем низкоразрядного квантования являются нейронные сети с двоичным весом (например, BinaryConnect (Courbariaux et al., 2015), BinaryNet (Courbariaux, Bengio, 2016), XNOR (Rastegari et al., 2016) и т. д.), где веса представлены с использованием только одного бита. Бинарные веса или активации обычно настраиваются непосредственно во время обучения сети с использованием определенных прямых и обратных правил. Например, BinaryConnect (Courbariaux et al., 2015) использует как детерминированную бинаризацию:

$$w_b = \text{sign}(w) = \begin{cases} +1, & \text{если } x \geq 0 \\ -1, & \text{в ином случае} \end{cases}, \quad (4.6)$$

так и стохастическую:

$$w_b = \begin{cases} +1, & \text{с вероятностью } p = \sigma(w) \\ -1, & \text{с вероятностью } 1 - p \end{cases}, \quad (4.7)$$

где σ – «жесткая» сигмоидная функция:

$$\sigma(x) = \max\left(0, \min\left(1, \frac{x+1}{2}\right)\right). \quad (4.8)$$

Это кусочно-линейная аппроксимация стандартной сигмоидной функции.

Схемы квантования

Для квантования с более высокой точностью (например, INT8) можно выполнить *квантование после настройки* (post-training quantization), когда веса и активации квантуются после настройки модели с полной точностью. Диапазон квантования для активаций определяется путем вычисления распределения на настроенном наборе, а слои пакетной нормализации (Ioffe, Szegedy, 2015) сворачиваются. Применение квантования INT8 после обуче-

ния обычно приводит к незначительной или нулевой потере точности¹. В недавней работе (Banner et al., 2019) изучалось квантование моделей до уровня INT4 после настройки.

Настройка с учетом квантования (quantization-aware training) может уменьшить потерю точности квантования за счет имитации квантования этапа вывода во время настройки (Jacob et al., 2017). На этапе настройки в сверточные слои вводится «ложный оператор квантования», а слои пакетной нормализации (Июффе и Сегеди, 2015) сворачиваются.

В случае как квантования после настройки, так и настройки с учетом квантования требуется прямой доступ к настроечным данным, чтобы получить хорошую точность после квантования, что не всегда возможно, особенно в приложениях, критичных в плане конфиденциальности. Для снижения битовой точности без доступа к настроечным данным применяется *квантование без данных*. Нагель и др. (Nagel et al., 2019) предложили выполнять квантование INT8 без данных, выравнивая диапазоны весов в сети. Сеть ZeroQ (Cai et al., 2020) оптимизирует набор дистиллированных данных, чтобы сопоставить статистику пакетной нормализации на разных уровнях сети для квантования без данных.

Низкоразрядная настройка

Помимо вывода, настройка с квантованными весами, активациями и градиентами может снизить затраты на обучение глубоких моделей. Настройка со смешанными 16-битными и 32-битными типами с плавающей запятой в модели широко поддерживается фреймворками глубокого обучения, такими как TensorFlow, PyTorch, TensorCores и т. д. Благодаря таким методам, как масштабирование потерь, настройка со смешанной точностью может снизить потребление памяти и повысить скорость настройки без потери точности. DoReFa-Net использует 1-битные веса, 2-битные активации и 6-битные градиенты для более быстрой настройки и вывода, что может обеспечить точность, сравнимую с 32-битной AlexNet (Крижевский и др., 2012) на ImageNet (Deng et al., 2009). В работе (Lin et al., 2015) стохастически бинаризуют веса, чтобы сократить время на умножение с плавающей запятой при настройке.

Аппаратная поддержка ускорения с низкой точностью

Квантованные модели могут уменьшить размер модели и объем памяти для развертывания, но для ускорения логического вывода им требуется аппаратная поддержка низкоразрядной арифметики. Квантование INT8 поддерживается мобильными процессорами ARM (например, Qualcomm Hexagon, ARM Neon), процессорами x86, графическими процессорами NVIDIA с TensorRT, FPGA Xilinx с DNNNDK и т. д. Сеть с двоичным квантованием также можно ускорить с помощью битовых операций. Более низкая битовая точность (например, троичная, INT4) в меньшей степени поддерживается на существующем оборудовании. Архитектура NVIDIA Turing поддерживает вывод INT4²,

¹ https://www.tensorflow.org/lite/performance/post_training_quantization.

² <https://developer.nvidia.com/blog/int4-for-ai-inference/>.

что обеспечивает дополнительное ускорение на 59 % по сравнению с INT8. Предпринимаются усилия по разработке специализированных аппаратных ускорителей для ускорения низкоразрядных квантованных моделей (Zhang et al., 2015; Sharify et al., 2018), которые обеспечивают превосходную энергоэффективность по сравнению с моделями полной точности.

В последнее время аппаратная поддержка квантования со смешанной точностью открывает новые возможности для повышения точности и снижения затрат. NVIDIA Turing Tensor Core поддерживает 1-битные, 4-битные, 8-битные и 16-битные арифметические операции; Imagination реализовала гибкую нейросетевую обработку изображений, которая поддерживает настройку битовой ширины для каждого слоя как для весов, так и для активаций. Недавно разработанный специализированный аппаратный ускоритель также обеспечивает поддержку смешанной точности: вычислительное оборудование, основанное на блоках умножения битовых последовательностей (Judd et al., 2016; Umuroglu et al., 2018), поддерживает временное умножение с разрядностью от 1 до 8 бит; BitFusion (Sharma et al., 2018) поддерживает пространственное умножение 2, 4, 8 и 16 бит.

4.1.4. Дистилляция знаний

Дистилляция знаний (knowledge distillation, KD; Bucilua et al., 2006; Hinton et al., 2015) может перенести так называемые «темные знания», полученные и сохраненные в «черном ящике» большой модели (учитель), в меньшую модель (ученик), чтобы объединить качество большой модели и скорость маленькой. Маленькая модель – это либо сжатая, либо более мелкая/узкая модель. Некоторые исследователи достигают цели, обучая меньшую сеть сопоставлять выходные логиты (Bucilua et al., 2006); Хинтон и др. (Hinton et al., 2015) предложили идею температуры softmax-выхода и обучили меньшую модель имитировать смягченное распределение softmax модели-учителя. KD демонстрирует многообещающие результаты в различных задачах классификации изображений, несмотря на простую реализацию.

Помимо конечных выходных логитов, полезную информацию также содержат промежуточные активации. Метод FitNet (Ромеро и др., 2014) обучает ученика имитировать полную карту характеристик модели учителя посредством регрессии. Метод *передачи внимания* (attention transfer, AT) (Загоруйко и Комодакис, 2016) передает карту внимания активации от учителя к ученику, используя суммирование карты признаков по измерению канала. Оба метода требуют промежуточной активации для совместного использования одного и того же пространственного разрешения, что ограничивает выбор архитектуры обучаемой модели.

Методы на основе KD также применимы к приложениям, выходящим за рамки классификации, таким как обнаружение объектов (Chen et al., 2017), семантическая сегментация (Liu et al., 2019a), языковое моделирование (Sanh et al., 2019), синтез изображений (Ли и др., 2020) и т. д.

4.1.5. Автоматическое сжатие модели

Методы сжатия моделей могут повысить эффективность развернутых моделей. Однако результат сжатия модели во многом зависит от гиперпараметров. Например, разные слои в глубоких сетях имеют разную пропускную способность и чувствительность (например, первый слой в CNN обычно очень чувствителен к прореживанию). Следовательно, мы должны применять разные коэффициенты прореживания для разных слоев сети, чтобы достичь оптимальной производительности. Пространство проектирования настолько велико, что человеческая эвристика обычно неоптимальна, а ручное сжатие модели требует много времени. С этой целью предлагается автоматическое сжатие модели, позволяющее найти правильную политику сжатия без участия человека.

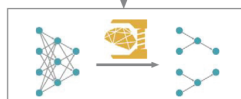
Автоматическое прореживание

Обычные методы сжатия моделей основаны на элементах, созданных вручную, и заставляют экспертов предметной области исследовать большое пространство проектирования, выбирая между размером модели, скоростью и точностью: обычно этот подход неоптимален и требует много времени. В методе AutoML для сжатия моделей (AutoML for Model Compression, AMC) (He et al., 2018) применяется обучение с подкреплением для эффективной выборки проектного пространства и поиска оптимальной политики сжатия сети (рис. 4.5). Мы обрабатываем предварительно обученную сеть (например, MobileNet-V1) послойно. Наш агент обучения с подкреплением DDPG (Lillicrap et al., 2015) получает представление s_t из слоя t и выводит коэффициент разреженности a_t . После того как слой сжат с коэффициентом a_t , агент переходит к следующему слою L_{t+1} . Затем оценивается точность модели, в которой сжаты все слои. Наконец, агенту обучения с подкреплением возвращается вознаграждение R , зависящее от точности модели и результата операции умножения-накопления (multiply-accumulate, MAC).

Сжатие модели человеком:
трудоемко, неоптимально



Движок AMC



Сжатие модели с помощью ИИ:
автоматическое, более высокая
степень сжатия, быстрее

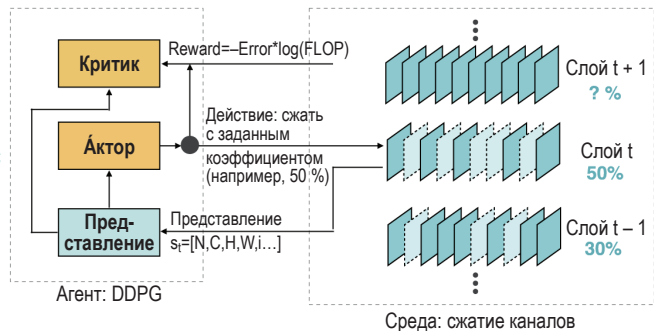


Рис. 4.5 ❖ Обзор механизма AutoML for Model Compression (AMC). Слева: AMC заменяет человека и делает сжатие модели полностью автоматизированным, при этом работая лучше, чем человек. Справа: формулировка AMC как задачи обучения с подкреплением (He et al., 2018)

При мелко модульном прореживании ResNet-50 (He et al., 2016) AMC может превзойти экспертов-людей в полностью автоматизированном режиме: этот движок повышает коэффициент сжатия ResNet-50 в ImageNet, настроенный экспертами, с 3,4 до 5 (см. рис. 4.6) без потери точности. AMC также может находить шаблоны прореживания, подобные человеческим эвристикам. Плотность каждого слоя на каждом этапе показана на рис. 4.7. Пики и гребни показывают, что RL-агент автоматически обучается сжимать сверточные слои 3×3 с большей разреженностью, поскольку они обычно имеют большую избыточность; в то же время у более компактных сверток 1×1 разреженность меньше. Статистика плотности каждого блока представлена на рис. 4.6. Видно, что распределение плотности AMC сильно отличается от распределения человека-эксперта, показанного в табл. 3.8 (Han, 2017). Мы предполагаем, что AMC может полностью исследовать рабочее пространство и лучше распределить разреженность.

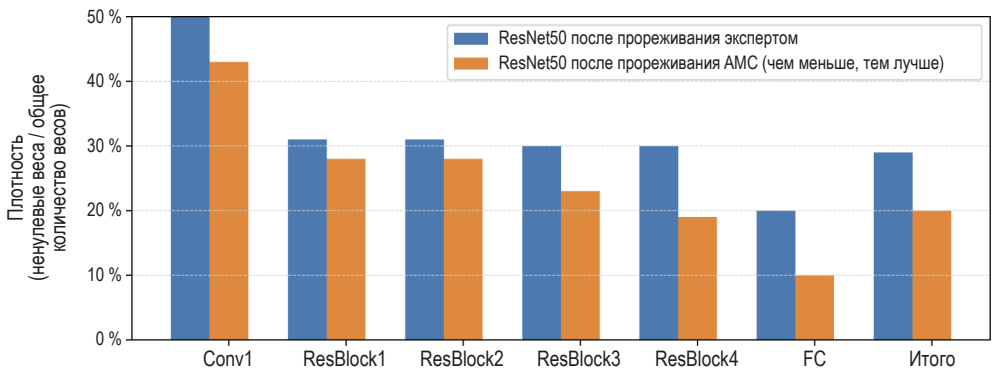


Рис. 4.6 ❖ AMC может сильнее сжать модель по сравнению с экспертами без потери точности (эксперт: сжатие ResNet50 в 3,4 раза; AMC: сжатие ResNet50 в 5 раз)

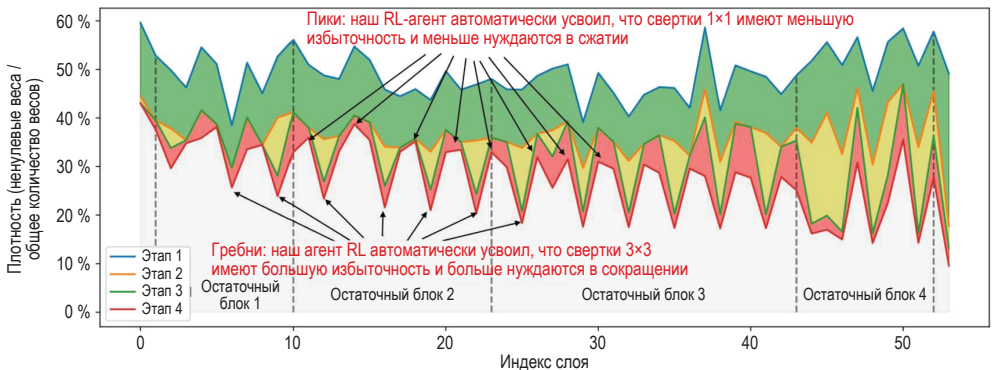


Рис. 4.7 ❖ Политика прореживания (коэффициент сжатия), заданная агентом AMC для ResNet-50. С помощью 4 этапов итеративного сжатия AMC находит очень выраженную картину распределения разреженности по слоям: пики представляют собой свертки 1×1 , гребни – свертки 3×3 . Агент обучения с подкреплением автоматически обнаруживает, что свертка 3×3 более избыточна, чем свертка 1×1 , и может быть больше прорежена

АМС влияет на аппаратное быстроедействие: он может оптимизировать не только объем вычислений (т. е. MAC), но и фактическую задержку на устройстве (рис. 4.8б). Мы используем очень компактную сеть MobileNetV1 (Howard et al., 2017) в качестве примера для измерения того, насколько мы можем улучшить скорость логического вывода. Предыдущие попытки использовать созданную *вручную* политику прореживания MobileNet-V1 привели к значительному снижению точности (Li et al., 2016b): прореживание исходных параметров MobileNet-V1 до 75,5 % приводит к точности 67,2 %¹, что даже хуже, чем исходные 75 % MobileNet-V1. Однако политика АМС значительно улучшает качество прореживания ImageNet, обеспечивая лучшую кривую Парето для точности при компромиссе по вычислениям (т. е. повышение точности при том же объеме вычислений). Как показано на рис. 4.8а, созданная вручную экспертом политика обеспечивает несколько худшую точность, чем исходная MobileNet-V1, при двукратном снижении MAC. Политика АМС также превосходит другую политику, основанную на эвристике (Yang et al., 2018), оптимально сочетая точность и быстроедействие.

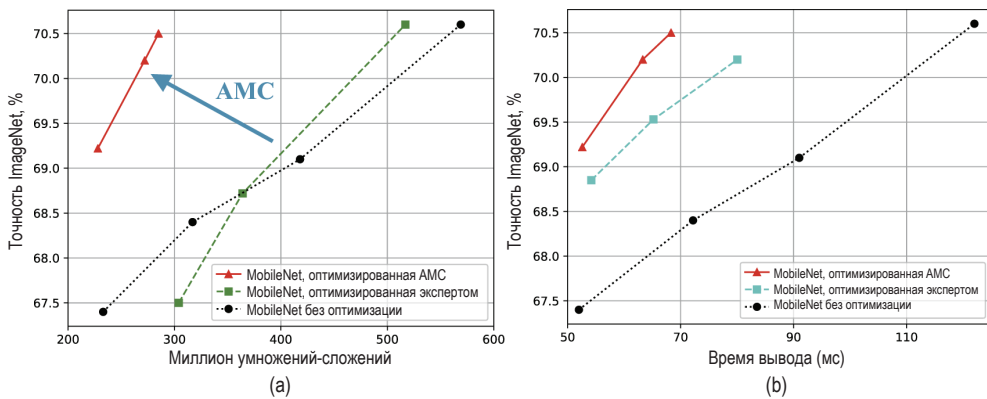


Рис. 4.8 ❖ (а) Сравнение точности и компромисса по MAC между АМС, экспертом и несжатой MobileNet-v1. Движок АМС явно доминирует над экспертом на оптимальной кривой Парето. (б) Сравнение компромисса между точностью и быстроедействием среди АМС, NetAdapt и MobileNet-V1 без сжатия. Движок АМС значительно улучшает кривую Парето для MobileNet-V1. Результат, полученный АМС на основе обучения с подкреплением, превосходит NetAdapt на основе эвристики по кривой Парето (время вывода измерено на смартфоне Google Pixel 1)

В недавно опубликованной работе (Liu et al., 2019) алгоритм MetaPruning сначала обучает PruningNet – своего рода метасеть, которая способна генерировать весовые параметры для любой прореженной структуры с учетом целевой сети, а затем использовать ее для поиска наилучшей политики прореживания при разных граничных условиях.

¹ <http://machinethink.net/blog/compressing-deep-neural-nets/>.

Автоматическое квантование

Реализация квантования со смешанной точностью также требует значительных усилий, направленных на определение оптимальной разрядности каждого слоя, чтобы достичь наилучшего соотношения между точностью и производительностью. Для автоматизации процесса предложено (Wang et al., 2018) использовать *аппаратно-зависимое автоматическое квантование* (hardware-aware automated quantization, HAQ) (рис. 4.9). Подход HAQ основан на обучении с подкреплением для автоматического определения политики квантования. Он использует обратную связь аппаратного ускорителя в цикле построения модели, а не полагается на опосредованные сигналы, такие как MAC и размер модели. По сравнению с обычными методами, HAQ полностью автоматизирован и может настраивать политику квантования для различных архитектур нейронных сетей и аппаратных архитектур.

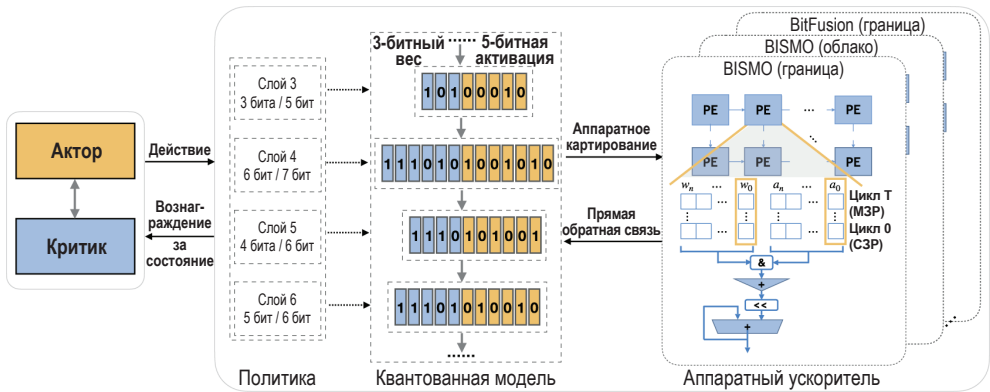


Рис. 4.9 ❖ Структурная схема метода HAQ. Данный метод использует обучение с подкреплением для автоматического итерационного поиска в огромном пространстве вариантов квантования с аппаратной обратной связью. Агент предлагает оптимальную политику распределения разрядностей с учетом количества вычислительных ресурсов (т. е. быстродействия, мощности и размера модели). RL-агент включает аппаратный ускоритель в цикл итерации, чтобы он мог получать прямую обратную связь от оборудования, вместо того чтобы полагаться на косвенные показатели (Wang et al., 2018)

HAQ использует совершенно разные политики квантования для *границных* (edge) и *облачных* (cloud) ускорителей. Политика квантования MobileNet-V1 на ускорителе BISMO (Umuroglu et al., 2018) (как пограничная, так и облачная конфигурация) представлена на рис. 4.10. На граничном ускорителе агент RL выделяет *меньше* битов активации для глубинных сверток в связи с тем, что такие свертки ограничены быстродействием памяти и активации и перегружают канал доступа к памяти. На облачном ускорителе наш агент выделяет *больше* битов глубинным сверткам и *меньше* битов точечным сверткам, поскольку облачное устройство имеет большую пропускную способность памяти и высокий параллелизм, поэтому сеть скорее испытывает вычислительные ограничения.

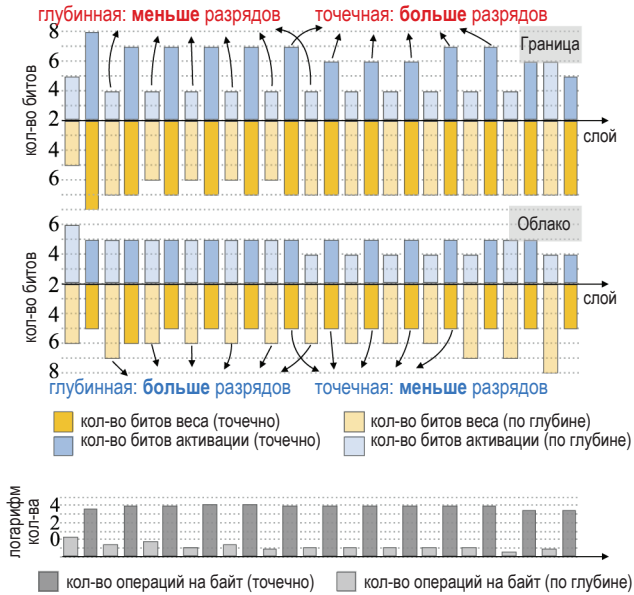


Рис. 4.10 ❖ Политика квантования при ограничениях по быстродействию для MobileNet-V1 (Wang et al., 2018)

4.2. ЭФФЕКТИВНЫЕ АРХИТЕКТУРЫ НЕЙРОННЫХ СЕТЕЙ

В дополнение к сжатию существующих глубоких нейронных сетей другим широко распространенным подходом к повышению эффективности является разработка новой архитектуры нейронной сети. Модель CNN обычно состоит из слоев свертки, слоев пулинга и полносвязных слоев, где большая часть вычислительной нагрузки исходит от слоев свертки. Например, в ResNet-50 (He et al., 2016) более 99 % операций умножения-накопления (MAC) выполняются для сверточных слоев. Следовательно, разработка эффективных слоев свертки является основой построения эффективных архитектур CNN.

В этом разделе сначала описывается стандартный сверточный слой, а затем описываются три эффективных варианта этого слоя. Далее мы представляем три варианта создаваемых вручную эффективных архитектур CNN, включая SqueezeNet (Iandola et al., 2016), MobileNets (Howard et al., 2017; Sandler et al., 2018) и ShuffleNets (Ma et al., 2018; Zhang et al., 2017). Наконец, мы описываем автоматизированные методы проектирования эффективных архитектур CNN.

4.2.1. Стандартный сверточный слой

Стандартный сверточный слой параметризуется ядром свертки K размера $O_c \times I_c \times K \times K$, где O_c – количество выходных каналов, I_c – количество входных каналов, K – пространственная размерность ядра (рис. 4.11а). Здесь для простоты мы предполагаем, что ширина и высота ядра свертки одинаковы. Также возможно иметь асимметричные ядра свертки (Szegedy, Vanhoucke, 2016).

Имея входную карту признаков F_i размера $I_c \times H \times W$, вычисляем выходную карту признаков F_o размера $O_c \times H \times W$ следующим образом¹:

$$F_o[n, h, w] = \sum_{m, i, j} K[n, m, i, j] \times F_i[m, h + 1 - \lfloor K/2 \rfloor + j - \lfloor K/2 \rfloor]. \quad (4.9)$$

Далее мы будем использовать выражение $F_o = \text{Conv}_{K \times K}(F_i, K)$ для представления стандартного слоя свертки с размером ядра K . Согласно уравнению (4.9), вычислительная стоимость стандартной свертки равна

$$\#MACs(\text{Conv}_{K \times K}) = H \times W \times O_c \times I_c \times K \times K, \quad (4.10)$$

в то время как количество параметров задается как

$$\#Params(\text{Conv}_{K \times K}) = O_c \times I_c \times K \times K. \quad (4.11)$$

4.2.2. Эффективные сверточные слои

Точечная свертка 1×1

Свертка 1×1 (также называемая точечной сверткой) – это особый вид стандартного сверточного слоя, где размер ядра K равен 1 (рис. 4.11d). Согласно уравнениям (4.10) и (4.11), замена стандартного слоя свертки $K \times K$ на слой свертки 1×1 уменьшит количество MAC ($\#MAC$) и количество параметров ($\#Params$) в K^2 раз. На практике, поскольку свертка 1×1 сама по себе не может агрегировать пространственную информацию, она комбинируется с другими сверточными слоями для формирования архитектуры CNN. Например, свертка 1×1 обычно используется для уменьшения/увеличения размерности канала карты признаков в CNN.

Групповая свертка

В отличие от свертки 1×1 , которая снижает вычислительную стоимость за счет уменьшения размерности ядра, групповая свертка снижает стоимость за счет уменьшения размерности канала. В частности, входную карту признаков F_i разбивают на G групп по измерению канала (рис. 4.11b):

$$\text{split}(F_i) = (F_i[0 : c, :, :], F_i[c : 2c, :, :], \dots, F_i[I_c - c : I_c, :, :]), \text{ где } c = I_c/G.$$

¹ Предполагая, что шаг равен 1, а заполнение нулями применяется для сохранения пространственного измерения карты объектов.

Затем каждую группу пропускают через стандартную свертку $K \times K$ размера $\frac{O_c}{G} \times \frac{O_c}{G} \times K \times K$. Наконец, выходные данные объединяются по измерению канала. По сравнению со стандартной сверткой $K \times K$, показатели #MAC и #Params уменьшаются при групповой свертке в G раз.

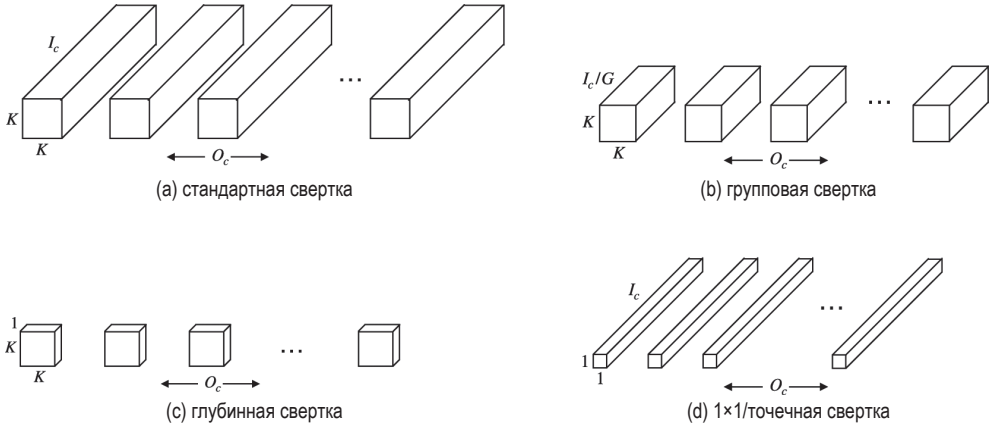


Рис. 4.11 ❖ Иллюстрация стандартной свертки и трех часто используемых эффективных вариантов

Глубинная свертка

В групповых свертках количество групп G является настраиваемым гиперпараметром. Чем больше G , тем меньше вычислительные затраты и меньше параметров. Крайним случаем является то, что G равно количеству входных каналов I_c . В этом случае слой групповой свертки называется *глубинной сверткой* (depthwise convolution) (рис. 4.11в). Значения #MAC и #Params глубинной свертки:

$$\#MACs(DWConv_{K \times K}) = H \times W \times O_c \times K \times K; \quad (4.12)$$

$$\#Params(DWConv_{K \times K}) = O_c \times K \times K, \quad (4.13)$$

где $O_c = I_c$.

4.2.3. Разработанные вручную эффективные модели CNN

SqueezeNet

Архитектура SqueezeNet (Iandola et al., 2016) (рис. 4.12) ориентирована на чрезвычайно компактные модели для мобильных приложений. Она имеет всего 1,2 млн параметров, но ее точность аналогична AlexNet (табл. 4.1). SqueezeNet имеет 26 слоев свертки и ни одного полносвязного слоя. Последняя карта признаков проходит через глобальный средний пулинг (global

average pooling) и образует вектор из 1000 измерений для подачи на слой softmax. SqueezeNet имеет восемь модулей Fire. Каждый модуль Fire содержит слой сжатия со сверткой 1×1 и парой сверток 1×1 и 3×3 . Модель SqueezeNet в формате Caffemodel¹ достигла точности top-1 57,4 % и top-5 80,5 % на наборе ImageNet 2012 (Deng et al., 2009)². SqueezeNet широко используется в мобильных приложениях, в которых размер модели жестко ограничен.

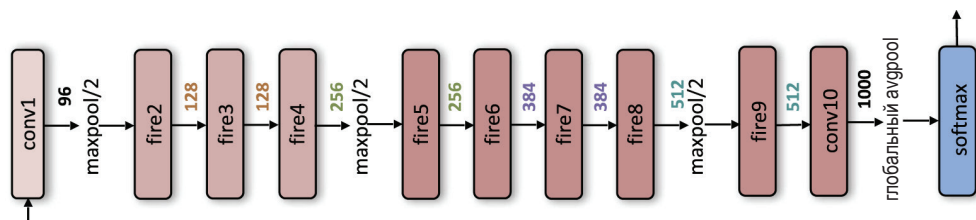


Рис. 4.12 ❖ Архитектура SqueezeNet (Iandola et al., 2016)

Таблица 4.1. Обобщенные результаты тестирования архитектур CNN, разработанных вручную, на наборе ImageNet

Сеть	#Params, млн	#MAC, млн	ImageNet	
			Top-1	Top-5
AlexNet (Krizhevsky et al., 2012)	60	720	57,2 %	80,3 %
GoogleNet (Szegedy et al., 2015)	6,8	1550	69,8 %	89,5 %
VGG-16 (Simonyan, Zisserman, 2014)	138	15300	71,5 %	–
ResNet-50 (He et al., 2016)	25,5	4100	76,1 %	92,9 %
SqueezeNet (Iandola et al., 2016)	1,2	1700	57,4 %	80,5 %
MobileNetV1 (Howard et al., 2017)	4,2	569	70,6 %	89,5 %
MobileNetV2 (Sandler et al., 2018)	3,4	300	72,0 %	–
MobileNetV2-1.4 (Sandler et al., 2018)	6,9	585	74,7 %	–
ShuffleNetV1-1.5x (Zhang et al., 2017)	3,4	292	71,5 %	–
ShuffleNetV2-1.5x (Ma et al., 2018)	3,5	299	72,6 %	–
ShuffleNetV2-2x (Ma et al., 2018)	7,4	591	74,9 %	–

Мобильные сети

Архитектура MobileNetV1 (Howard et al., 2017) основана на структурном блоке, называемом *сверткой с разделением по глубине* (depthwise separable convolution) (рис. 4.13а), который состоит из слоя глубинной свертки 3×3 и слоя свертки 1×1 . Входное изображение сначала проходит через стандартный слой

¹ Caffe – среда для глубокого обучения, разработанная Яньцинем Цзя (Yangqing Jia). Название Caffe произошло от сокращения «Convolution Architecture For Feature Extraction» (Сверточная архитектура для извлечения признаков). <https://ru.wikipedia.org/wiki/Caffe>. – Прим. перев.

² Критерий top-1 считает ответ верным только тогда, когда наиболее вероятный ответ модели совпадает с правильным; top-5 означает, что правильный ответ попал в один из пяти наиболее вероятных ответов. – Прим. перев.

свертки 3×3 со страйдом 2, затем через 13 блоков свертки с разделением по глубине. Наконец, карта признаков проходит через глобальный средний пулинг и образует вектор из 1280 измерений, который передается на последний полносвязный слой с 1000 выходных блоков (output unit). Обладая 569 млн MAC и 4,2 млн параметров, модель MobileNetV1 достигла точности top-170,6 % на наборе ImageNet 2012 (табл. 4.1).

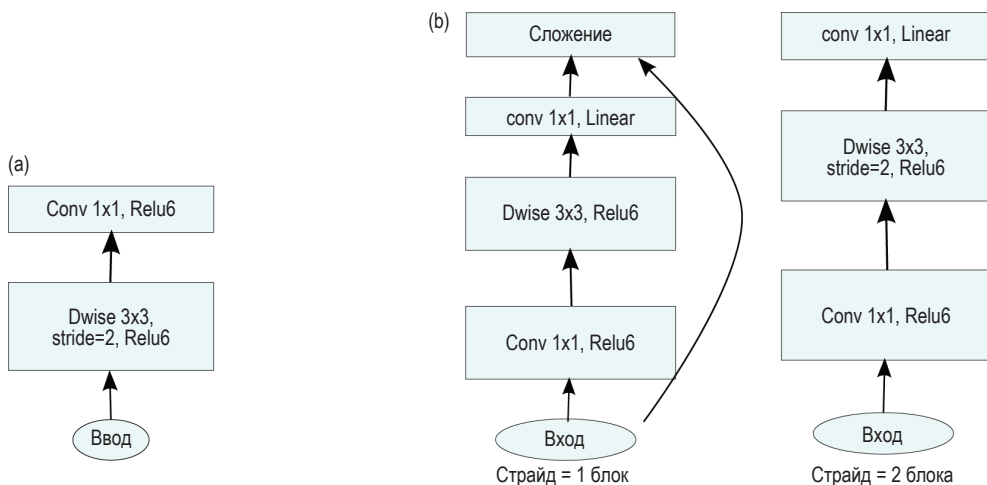


Рис. 4.13 ❖ (a) Структурный блок MobileNetV1 (Howard et al., 2017). Он состоит из слоя глубинной свертки 3×3 и слоя свертки 1×1 . (b) Структурные блоки MobileNetV2 (Sandler et al., 2018). Каждый блок состоит из слоя глубинной свертки 3×3 и двух слоев свертки 1×1 . Когда страйд равен 1, блок выполняет сквозное пропускание «вход \rightarrow выход»

MobileNetV2 (Sandler et al., 2018), улучшенная версия MobileNetV1, также использует в своих структурных блоках глубинные свертки 3×3 и свертки 1×1 . В отличие от MobileNetV1, структурный блок в MobileNetV2 состоит из трех слоев, включая слой свертки 3×3 и два слоя свертки 1×1 (рис. 4.13). Интуитивно понятно, что мощность глубинной свертки намного ниже, чем у стандартной свертки, и поэтому для повышения ее производительности требуется больше каналов. С точки зрения затрат, по мере увеличения количества каналов показатели #MAC и #Params глубинной свертки растут только линейно, а не квадратично, как в случае стандартной свертки. Даже при большом количестве каналов стоимость слоя свертки по глубине остается умеренной. Таким образом, в MobileNetV2 входная карта объектов сначала проходит свертку 1×1 , чтобы увеличить размер канала на коэффициент, называемый *коэффициентом расширения* (expand ratio). Затем расширенная карта признаков подается на свертку по глубине 3×3 , за которой следует еще одна свертка 1×1 , чтобы уменьшить размер канала до исходного значения. Эта структура называется *перевернутым узким местом* (inverted bottleneck), а блок называется *мобильным перевернутым узким местом* (mobile inverted bottleneck). Помимо перевернутого мобильного узкого места,

MobileNetV2 имеет еще два усовершенствования по сравнению с MobileNetV1. Во-первых, MobileNetV2 обеспечивает сквозной канал¹ для блоков, в которых шаг равен 1. Во-вторых, удаляется функция активации последней свертки 1×1 в каждом блоке. Сочетая эти улучшения, MobileNetV2 достигает 72,0 % точности первого уровня в ImageNet 2012 всего с 300 млн MAC-адресов и 3,4 млн параметров (табл. 4.1).

ShuffleNet

Подобно MobileNets, ShuffleNetV1 использует глубинную свертку 3×3 , а не стандартную свертку. Кроме того, в ShuffleNetV1 представлены две новые операции: *точечная групповая свертка* (pointwise group convolution) и *перетасовка каналов* (channel shuffle). Идея точечной групповой свертки заключается в снижении вычислительных затрат на слои свертки 1×1 . Однако у нее есть побочный эффект: группа не может видеть информацию от других групп. Это значительно вредит точности. Для устранения данного побочного эффекта введена операция перетасовки каналов путем обмена картами признаков между различными группами. Операция перетасовки каналов изображена на рис. 4.14. После перетасовки каждая группа будет содержать информацию из всех групп. В ImageNet 2012 ShuffleNetV1 достигает точности 71,5 % в top-1 с 292 млн MAC (табл. 4.1).

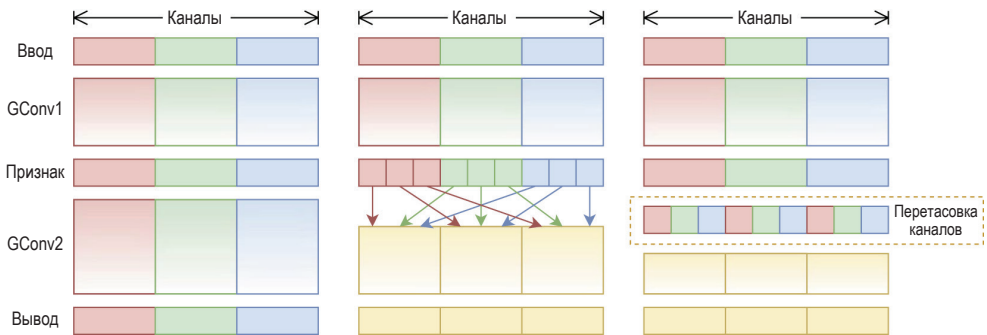


Рис. 14.14 ❖ Иллюстрация операции перетасовки каналов (Zhang et al., 2017)

В ShuffleNetV2 входная карта признаков разделена на две группы в начале каждого стандартного блока. Одна группа проходит через ветвь свертки, состоящую из слоя глубинной свертки 3×3 и двух слоев свертки 1×1 . Другая группа проходит через сквозной канал, когда страйд равен 1, и проходит свертку 3×3 с разделением по глубине, когда страйд равен 2. В конце выходные данные объединяются по измерению канала, после чего следует операция перетасовки каналов для обмена информацией между группами. При 299 млн MAC ShuffleNetV2 достигает точности top-1 72,6 % в ImageNet 2012 (табл. 4.1).

¹ При сквозном канале $\text{вывод} = F(\text{ввод}) + \text{ввод}$. Без сквозного канала $\text{вывод} = F(\text{ввод})$.

4.2.4. Поиск нейронной архитектуры

Успех вышеупомянутых эффективных моделей CNN зависит от созданных вручную архитектур нейронных сетей, которые требуют, чтобы эксперты в предметной области исследовали большое пространство проектирования, находя компромисс между размером модели, задержкой, энергией и точностью. Это очень трудоемкий процесс, который, как правило, дает не самую оптимальную архитектуру. Как следствие растет интерес к разработке автоматизированных методов для решения этой задачи.

Поиск нейронной архитектуры (neural architecture search, NAS) представляет собой использование методов машинного обучения для автоматического проектирования архитектуры нейронной сети. В традиционной формулировке NAS (Zoph, Le, 2016) проектирование архитектур нейронных сетей моделируется как задача генерации последовательности, где для создания архитектур нейронных сетей вводится авторегрессионный контроллер RNN. Этот контроллер обучается путем многократного выбора архитектур нейронной сети, их оценивания и обновления контроллера на основе обратной связи. Чтобы найти хорошую архитектуру нейронной сети в огромном пространстве поиска, этот процесс обычно должен обучить для целевой задачи и оценить десятки тысяч нейронных сетей, что приводит к неприемлемо высокой вычислительной нагрузке (10^4 часов GPU). Для решения данной проблемы предлагается множество методов, направленных на улучшение различных компонентов NAS, включая пространство поиска, алгоритм поиска и стратегию оценки качества модели.

Пространство поиска

Для всех методов NAS требуется предопределенное пространство поиска, содержащее основные элементы сети и их взаимные соединения. Например, типичные базовые элементы моделей CNN состоят из (а) сверток (Zoph et al., 2017; Real et al., 2018): стандартных сверток (1×1 , 3×3 , 5×5), асимметричных сверток (1×3 и 3×1 , 1×7 и 7×1), сверток с разделением по глубине (3×3 , 5×5), расширенных сверток (3×3); (б) пулинга: average-пулинга (3×3), max-пулинга (3×3); (в) функции активации (Ramachandran et al., 2017). Затем выполняется последовательная укладка этих элементов (Baker et al., 2016) с тождественными связями (Zoph and Le, 2016). Полное пространство поиска на сетевом уровне экспоненциально растет по мере углубления сети (рис. 4.15а). Когда глубина равна 20, это пространство поиска содержит более 10^{36} различных архитектур нейронных сетей (Zoph et al., 2017).

Очень эффективным подходом к повышению скорости поиска является ограничение рабочего пространства. В частности, в некоторых работах (Zoph et al., 2017 г.) Чжун и др. (2017) предлагают искать не всю архитектуру нейронной сети, а базовые структурные ячейки (рис. 4.15б), которые можно складывать для построения нейронных сетей. При таком подходе сложность архитектуры не зависит от глубины сети, а обученные ячейки можно передавать из разных наборов данных. Это позволяет NAS выполнять поиск в небольшом наборе прокси-данных (например, CIFAR-10), а затем переходить

к другому крупномасштабному набору данных (например, ImageNet), подбирая количество ячеек. Внутри ячейки сложность еще больше снижается за счет поддержки иерархических топологий (Liu et al., 2018a) или постепенного увеличения количества элементов (от простого к сложному) (Liu et al., 2017).

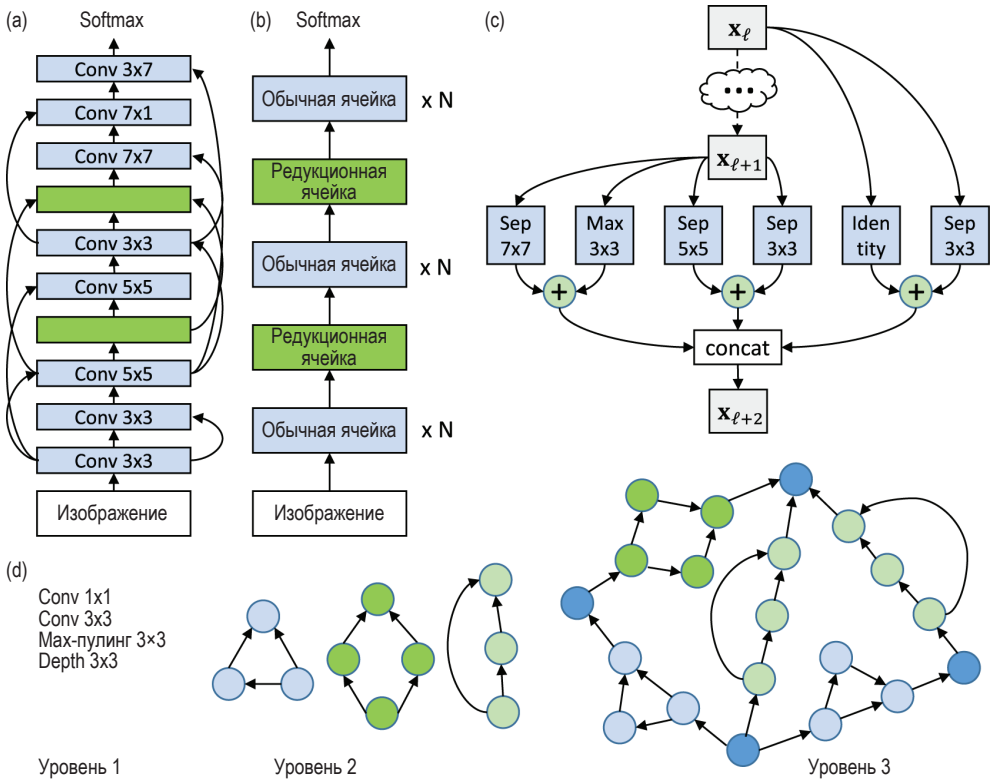


Рис. 4.15 ❖ Пространство поиска NAS (Deng et al., 2020): (a) пространство поиска на уровне сети (Zoph and Le, 2016); (b) пространство поиска на уровне ячеек (Zoph et al., 2017); (c) пример изученной структуры ячеек (Liu et al., 2017); (d) трехуровневое иерархическое пространство поиска (Liu et al., 2018a)

Алгоритм поиска

В методах NAS каждый шаг поиска обычно разбит на два этапа: (1) генератор создает архитектуру, а затем (2) оценщик обучает сеть и измеряет ее показатели. Поскольку оценка качества архитектуры требует полноценного обучения получившейся нейронной сети, что очень дорого, алгоритмы поиска, влияющие на эффективность выборки, играют важную роль в повышении скорости поиска NAS. Большинство алгоритмов поиска, используемых в NAS, делятся на пять категорий: случайный поиск, обучение с подкреплением (RL), эволюционные алгоритмы, байесовская оптимизация и методы на основе градиента. Среди них RL, эволюционные алгоритмы и методы на основе градиента обеспечивают наиболее конкурентоспособные результаты.

Методы на основе RL моделируют процесс генерации архитектуры как марковский процесс принятия решений, рассматривают точность модели выбранной архитектуры как вознаграждение и обновляют модель генератора архитектуры с использованием алгоритмов RL, включая Q-обучение (Baker et al., 2016; Zhong et al., 2017), REINFORCE (Zoph and Le, 2016), PPO (Zoph et al., 2017) и т. д. Иногда вместо обучения модели генератора архитектуры используются эволюционные методы (Real et al., 2018; Liu et al., 2018), поддерживающие популяцию архитектур нейронных сетей. Эта популяция обновляется посредством мутаций и рекомбинаций. В то время как методы на основе RL и эволюционные методы оптимизируют архитектуру нейронных сетей в дискретном пространстве, DARTS (Liu et al., 2018b) предлагает непрерывное представление архитектуры:

$$y = \sum_i \alpha_i o_i(x), \quad \text{где } \alpha_i \geq 0, \sum_i \alpha_i = 1, \quad (4.14)$$

где $\{\alpha_i\}$ обозначают параметры архитектуры, o_i – операции-кандидаты, x – входные данные, а y – выходные данные. Данный подход позволяет оптимизировать архитектуры нейронных сетей в непрерывном пространстве с помощью градиентного спуска, что значительно повышает эффективность поиска. Помимо вышеперечисленных методов, эффективность поиска NAS можно повысить, исследуя пространство архитектур с помощью операций преобразования сети, начиная с существующей сети и повторно используя веса (Cai et al., 2018a,b; Elsken et al., 2018).

Оценка качества модели

Чтобы направлять процесс поиска, методы NAS должны оценивать показатели качества (обычно точность на проверочном наборе) исследуемых нейронных архитектур. Тривиальный подход к получению этих характеристик заключается в обучении выбранных архитектур нейронных сетей на обучающих данных и измерении их точности на проверочном наборе. Однако это ведет к чрезмерным вычислительным затратам (Zoph, Le, 2016; Zoph et al., 2017; Real et al., 2018). Поэтому разработаны новые методы, направленные на ускорение этапа оценки производительности.

В качестве альтернативы этап оценки можно ускорить с помощью гиперсети (Brock et al., 2017), которая может напрямую генерировать веса нейронной архитектуры без ее обучения. Хотя при использовании сгенерированных весов точность модели значительно ухудшится, эту точность можно использовать в качестве прокси-метрики для выбора нейронных архитектур. Таким образом, необходимо обучить только одну гиперсеть, что значительно снижает затраты на поиск. Точно так же методы One-shot NAS (Pham et al., 2018; Liu et al., 2018b; Cai et al., 2019) сосредоточены на обучении одной суперсети, от которой небольшие подсети напрямую наследуют веса без затрат на обучение.

Автоматическое проектирование или ручная разработка?

На рис. 4.16 представлены сводные показатели автоматически спроектированных моделей CNN в сравнении с моделями CNN, разработанными чело-

веком, после тестирования на наборе ImageNet. NAS не только экономит трудозатраты инженеров, но и обеспечивает лучшие модели CNN по сравнению с ручной разработкой. Помимо классификации ImageNet, автоматически спроектированные модели CNN превзошли модели, разработанные вручную, на задачах по обнаружению объектов (Zoph et al., 2017; Chen et al., 2019b; Ghiasi et al., 2019; Tan et al., 2020a) и семантической сегментации (Liu et al., 2019; Chen et al., 2018).

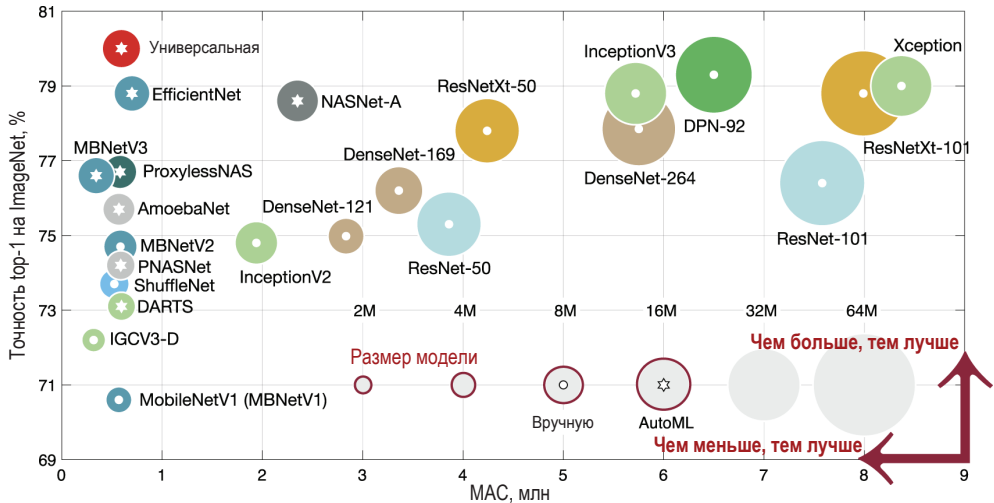


Рис. 4.16 ❖ Обобщенные результаты тестирования автоматически спроектированных моделей CNN и моделей CNN, разработанных вручную, на наборе ImageNet (Cai et al., 2020a)

4.2.5. Поиск нейронной архитектуры, ориентированной на оборудование

Хотя NAS продемонстрировал многообещающие результаты, добившись значительного снижения показателя MAC без ущерба для точности, в прикладных приложениях нас интересует реальная эффективность оборудования (например, задержка вывода, энергопотребление), а не число MAC. К сожалению, MAC-эффективность напрямую не связана с реальной аппаратной эффективностью. На рис. 4.17 приведено сравнение автоматически разработанных моделей CNN (NASNet-A и AmoebaNet-A) и моделей CNN, разработанных человеком (MobileNetV2-1.4). Хотя NASNet-A и AmoebaNet-A имеют меньший формальный показатель MAC, чем MobileNetV2-1.4, на самом деле они работают медленнее, чем MobileNetV2-1.4 на реальном оборудовании. Это связано с тем, что показатель MAC отражает только вычислительную сложность операций свертки. Другие факторы, такие как стоимость доступа к данным, параллелизм и стоимость поэлементных операций, которые существенно влияют на реальную эффективность оборудования, не учитываются.

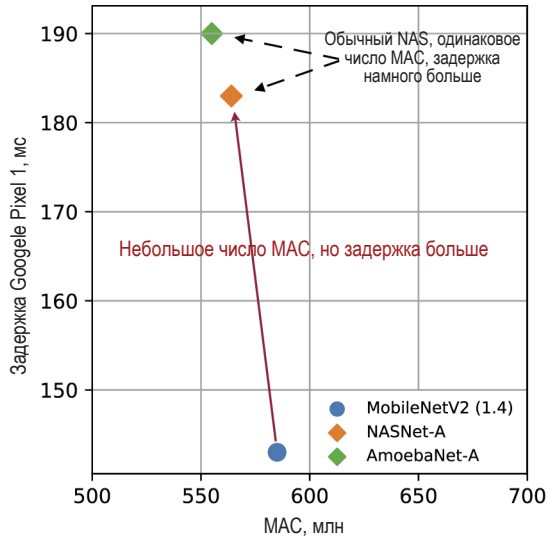


Рис. 4.17 ❖ Показатель MAC не отражает реальной эффективности оборудования. NASNet-A и AmoebaNet-A (модели CNN, разработанные автоматически) имеют меньший MAC, чем MobileNetV2-1.4 (модель CNN, разработанная человеком). Однако они работают медленнее, чем MobileNetV2-1.4 на Google Pixel 1

Эта проблема побуждает исследователей разрабатывать методы NAS с учетом особенностей аппаратного обеспечения (Tan et al., 2018; Cai et al., 2019; Wu et al., 2019), которые напрямую включают обратную связь с оборудованием в процесс поиска архитектуры. Пример аппаратного фреймворка NAS изображен на рис. 4.18. Помимо оценки точности, каждая выбранная архитектура нейронной сети проверяется на реальном оборудовании для сбора информации о задержках. Многоцелевое вознаграждение REW определяется на основе точности ACC и задержки LAT :

$$\text{reward} = ACC \times \left(\frac{LAT}{T} \right)^\omega, \quad (4.15)$$

где T – задержка на целевом оборудовании, а ω – гиперпараметр.

Прогнозирование задержки

Измерение задержки на реальном устройстве является точным, но не идеальным способом поиска масштабируемой нейронной архитектуры. На то есть две причины: (а) *низкая скорость*. Например, в TensorFlow-Lite¹ нам нужно усреднить сотни прогонов, чтобы получить точное измерение, что занимает примерно 20 секунд на одну итерацию поиска архитектуры. Это намного медленнее, чем однократное выполнение прогона вперед/назад. (б) *Высокая стоимость*. Для создания автоматического конвейера сбора данных о задержках с мобильной фермы необходимо соединить в кластер

¹ <https://www.tensorflow.org/lite>.

большое количество физических мобильных устройств и разработать специальное программное обеспечение.

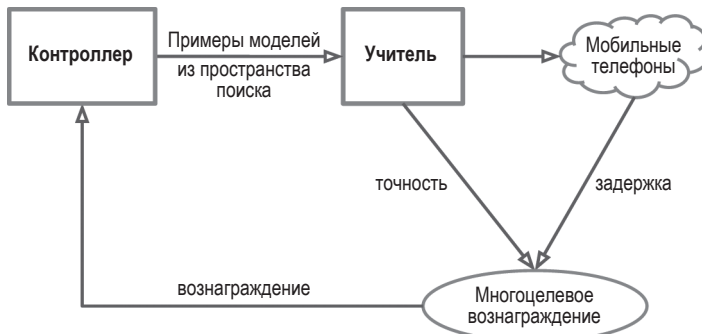


Рис. 4.18 ❖ Пример аппаратно-ориентированного алгоритма NAS (Tan et al., 2018)

По сравнению с прямым измерением более экономичным решением является создание модели прогнозирования для оценки задержки (Cai et al., 2019). На практике этот метод реализуется путем выборки архитектур нейронных сетей из пространства кандидатов и профилирования их задержки на целевой аппаратной платформе. Собранные данные затем используются для построения модели прогнозирования задержки. Для аппаратных платформ с последовательным выполнением операций, таких как мобильные устройства и FPGA, простой таблицы поиска по задержкам, которая сопоставляет каждую операцию с ее предполагаемой задержкой, достаточно, чтобы обеспечить очень точные прогнозы задержки (рис. 4.19). Еще одним

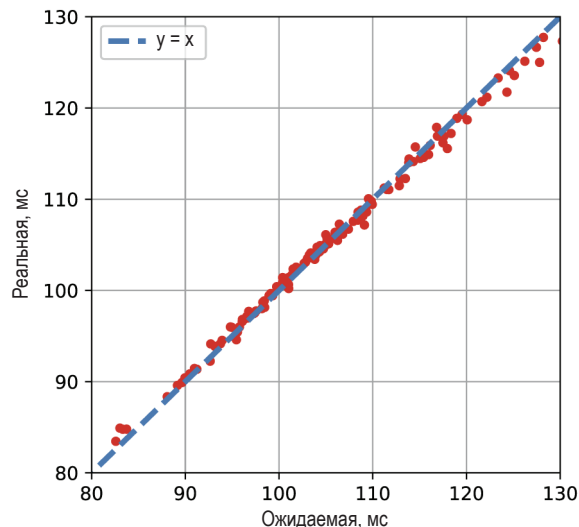


Рис. 4.19 ❖ Прогнозируемая задержка по сравнению с реальной задержкой в Google Pixel 1 (Cai et al., 2019)

преимуществом этого подхода является то, что он позволяет моделировать задержку нейронной сети как потерю регуляризации (рис. 4.20), позволяя оптимизировать компромисс между точностью и задержкой дифференцированным образом.

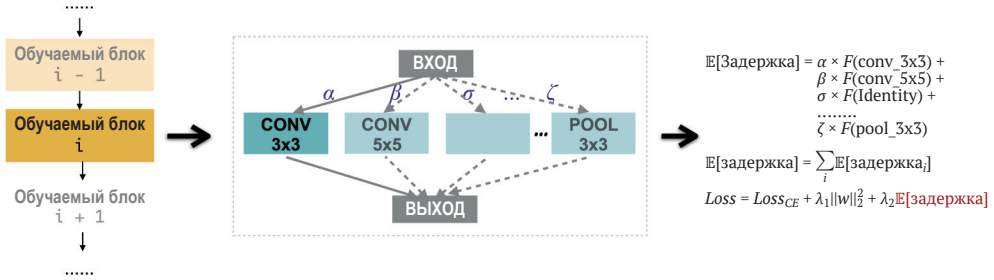


Рис. 4.20 ❖ Обеспечение дифференцируемости задержки путем введения потери задержки (Cai et al., 2019)

Специализированные модели для различного оборудования

Поскольку стоимость создания новой модели нейронной сети весьма высока, часто применяется развертывание одной и той же модели для всех аппаратных платформ. Однако это неоптимально, так как разные аппаратные платформы имеют разные свойства, такие как количество арифметических блоков, пропускная способность памяти, размер кеша и т. д. Используя аппаратные технологии NAS, можно получить специализированную архитектуру нейронной сети для каждого варианта оборудования.

На рис. 4.21 детально изображены архитектуры специализированных моделей CNN для графических процессоров и мобильных устройств. Мы заметили, что архитектуре присуще разное строение при ориентации на разные платформы: (а) модель для GPU более мелкая и широкая, особенно на ранних стадиях, когда карта признаков имеет более высокое разрешение; (б) модель GPU предпочитает большие операции MBConv (например, MBConv6 7×7), в то время как мобильная модель предпочитает меньшие операции MBConv. Это связано с тем, что GPU имеет гораздо более высокий параллелизм, чем процессор мобильного устройства, поэтому он может использовать преимущества больших операций MBConv. Еще одно интересное наблюдение заключается в том, что искомые модели на всех платформах предпочитают более крупные операции MBConv в первом блоке на каждом этапе, где карта признаков подвергается субдискретизации. Это может быть связано с тем, что более крупные операции MBConv выгодны для сети, так как они сохраняют больше информации при понижении разрешения.

В табл. 4.2 показаны сводные результаты специализированных моделей для графических процессоров и мобильных устройств. Интерес вызывает тот факт, что модели, оптимизированные для GPU, не работают быстро на мобильных устройствах, и наоборот. Поэтому важно сгенерировать специализированные нейронные сети для разных аппаратных архитектур, чтобы добиться максимальной эффективности на различном оборудовании.

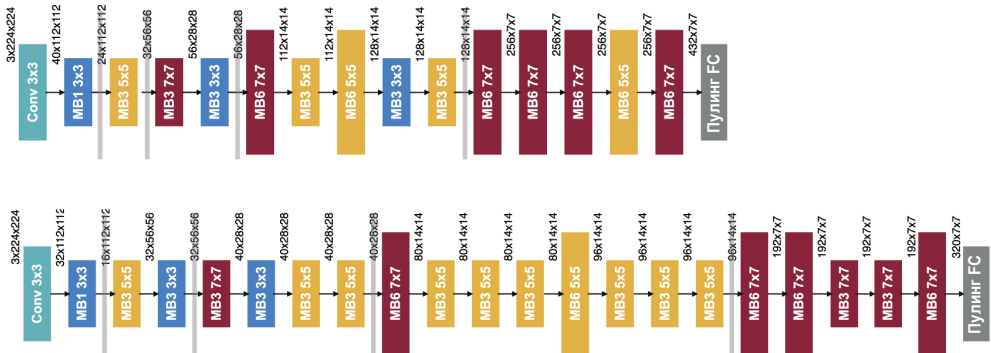


Рис. 4.21 ❖ Эффективные модели, оптимизированные для различного оборудования. MBConv3 и MBConv6 обозначают мобильный блок перевернутого узкого места с коэффициентом расширения 3 и 6 соответственно. Выводы: для GPU лучше подходят неглубокие и широкие модели с ранним пулингом; для мобильного оборудования – глубокие и узкие модели с поздним пулингом. Слои пулинга предпочитают большое и широкое ядро. Ранние слои предпочитают мелкие ядра. Поздние слои предпочитают крупные ядра (Cai et al., 2019)

Таблица 4.2. На реальном оборудовании лучше работают специализированные модели (Cai et al., 2019). При условии равной точности специализированная модель (ProxylessNAS-Mobile) снижает задержку в 1,8 раза по сравнению с неспециализированной моделью CNN (MobileNetV2-1.4). Кроме того, модели, оптимизированные для GPU, не так быстро работают на мобильных устройствах, и наоборот

Сеть	ImageNet Top-1, %	Задержка GPU, мс	Задержка мобильного устройства, мс
MobileNetV2-1.4 (Sandler et al., 2018)	74,7	—	143
ProxylessNAS-GPU (Cai et al., 2019)	75,1	5,1	124
ProxylessNAS-Mobile (Cai et al., 2019)	74,6	7,2	78

Поддержка нескольких аппаратных платформ и ограничения эффективности

Хотя специализированные модели CNN превосходят неспециализированные аналоги, разработка специализированных CNN для каждого сценария применения по-прежнему представляет собой трудную задачу как при ручном проектировании, так и с помощью аппаратных NAS, поскольку подобные методы требуют повторения процесса проектирования сети и переобучения спроектированной сети с нуля. Их совокупная стоимость растет линейно по мере увеличения количества сценариев развертывания, что приведет к избыточному потреблению энергии и выбросу CO₂ (Strubell et al., 2019). Это лишает сети возможности работать с огромным количеством оборудования (23,14 млрд устройств IoT в 2018 г.)¹ и высокодинамичными средами раз-

¹ <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>.

вертывания (различные параметры батареи питания, различные требования к задержке и т. д.).

Одним из многообещающих решений этой проблемы является создание универсальной сети (once-for-all, OFA) (Cai et al., 2020a; Yu et al., 2020), которую можно напрямую развернуть в различных архитектурных конфигурациях, сокращая затраты на обучение. Вывод выполняется путем выбора только части сети OFA. Архитектура гибко поддерживает различную глубину, ширину, размер ядра и разрешение без переобучения. Пример OFA изображен на рис. 4.22 (слева). В частности, этап обучения модели отделен от этапа поиска нейронной архитектуры. На этапе обучения модели основное внимание уделяется повышению точности всех подсетей, полученных путем выбора различных частей сети OFA. Подмножество подсетей выбирается на этапе специализации модели для обучения предикторов точности и предикторов задержки. Поиск архитектуры на основе предикторов (Liu et al., 2018) для получения специализированной подсети проводится с учетом целевого оборудования и ограничений, а стоимость незначительна¹. Таким образом, общая стоимость проектирования специализированной нейронной сети снижается с $O(N)$ до $O(1)$ (рис. 4.22, середина).

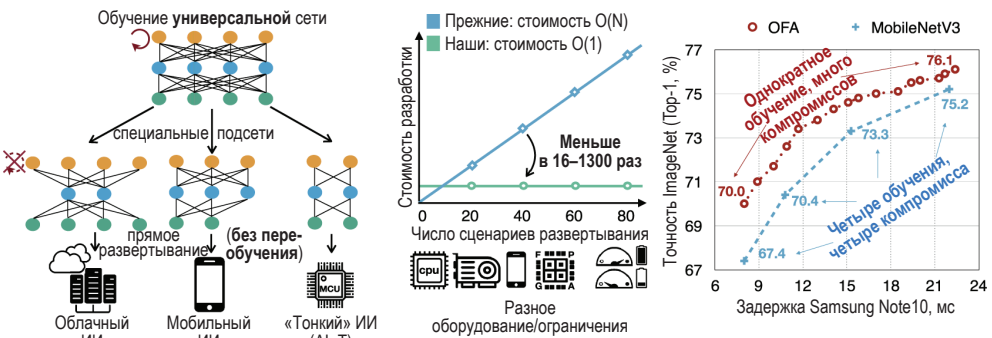


Рис. 4.22 ❖ Слева: единая универсальная сеть обучена поддерживать различные архитектурные конфигурации, включая глубину, ширину, размер ядра и разрешение. Специализированная подсеть выбирается напрямую из единой сети без обучения, с учетом сценария развертывания. Середина: этот подход снижает стоимость специализированного развертывания глубокого обучения с $O(N)$ до $O(1)$. Справа: универсальная сеть с последующим выбором модели может обеспечить множество компромиссов между точностью и задержкой путем обучения только один раз по сравнению с обычными методами, требующими повторного обучения (Cai et al., 2020)

В табл. 4.3 представлено сравнение методов OFA и современных аппаратных NAS на мобильном телефоне (Google Pixel 1). Стоимость OFA *постоянна*, в то время как стоимость других моделей *линейно* зависит от количества сценариев развертывания (N). При $N = 40$ общие выбросы CO_2 от OFA в 16 раз

¹ <https://github.com/mit-han-lab/once-for-all/blob/master/tutorial/ofa.ipynb>.

меньше, чем у ProxylessNAS, в 19 раз меньше, чем у FBNet, и в 1300 раз меньше, чем у MnasNet.

Таблица 4.3. Обобщенные результаты для телефона Pixel 1 (Cai et al., 2020). Первая группа соответствует разработанным человеком моделям CNN. Вторая группа – обычным NAS. Третья группа – аппаратно-ориентированным NAS. Последняя группа соответствует OFA. Метка «#75» означает, что специализированные подсети точно настроены на 75 эпох после захвата весов из сети OFA. «CO₂e» обозначает выбросы CO₂, которые рассчитываются на основе данных (Strubell et al., 2019). Стоимость AWS рассчитывается на основе стоимости крупных экземпляров Amazon AWS P3.16x, нагружаемых по требованию

Сеть	ImageNet top-1, %	MAC, млн	Мобильная задержка, мс	Стоимость (часов GPU)	Стоимость обучения (часов GPU)	Совокупная стоимость (N = 40)		
						Часов GPU, тыс.	CO ₂ e, тыс. фунтов	Стоимость AWS, тыс. долл.
MobileNetV2 (Sandler et al., 2018)	72,0	300	66	0	1507N	6	1,7	18,4
MobileNetV2 #1200	73,5	300	66	0	12007N	48	13,6	146,9
NASNet-A (Zoph et al., 2017)	74,0	564	–	480007N	–	1920	544,5	5875,2
DARTS (Liu et al., 2018b)	73,1	595	–	967N	2507N	14	4,0	42,8
MnasNet (Tan et al., 2018)	74,0	317	70	40000N	–	1600	453,8	4896,0
FBNet-C (Wu et al., 2019)	74,9	375	–	2167N	3607N	23	6,5	70,4
ProxylessNAS (Cai et al., 2019)	74,6	320	71	2007N	3007N	20	5,7	61,2
SinglePathNAS (Guo et al., 2019)	74,7	328	–	288 + 247N	3847N	17	4,8	52,0
AutoSlim (Yu, Autoslim, 2019)	74,2	305	63	180	3007N	12	3,4	36,7
MobileNetV3-Large (Howard et al., 2019)	75,2	219	58	–	1807N	7,2	1,8	22,2
OFA	76,0	230	58	40	1200	1,2	0,34	3,7
OFA #75	76,9	230	58	40	1200 + 757N	4,2	1,2	13,0
OFA _{Large} #75	80,0	595	–	40	1200 + 757N	4,2	1,2	13,0

На рис. 4.23 обобщены результаты OFA при разных значениях MAC и ограничениях задержки Pixel 1. Интересное наблюдение заключается в том, что обучение искомым нейронных архитектур с нуля не может достичь того же уровня точности, что и OFA, что позволяет предположить, что превосходным характеристикам OFA способствуют не только нейронные архитектуры, но и предварительно обученные веса.

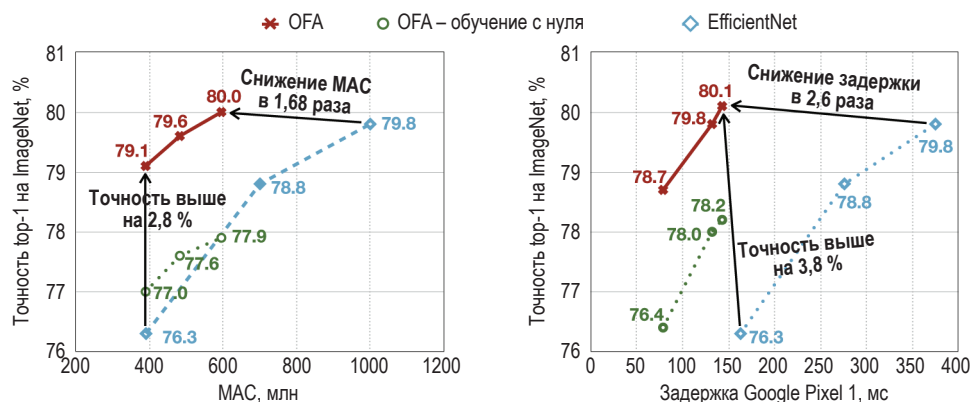


Рис. 4.23 ❖ Обучение нейронных архитектур поиска с нуля не может обеспечить такую же точность, как OFA (Cai et al., 2020)

4.3. ЗАКЛЮЧЕНИЕ

За последние несколько лет глубокие нейронные сети добились беспрецедентного успеха в области искусственного интеллекта; однако столь превосходные результаты достигаются за счет высокой вычислительной сложности. Это ограничивает их применение на многих периферийных устройствах, где аппаратные ресурсы сильно ограничены размерами корпуса, емкостью батареи и отсутствием системы охлаждения.

В этой главе представлен систематический обзор эффективных моделей глубокого обучения, позволяющий как исследователям, так и практикам быстро начать работу в данной области. Сначала мы описываем различные подходы к сжатию моделей, ставшие отраслевыми стандартами, такие как прореживание, факторизация, квантование и эффективное проектирование моделей. Далее описываем новые подходы, направленные на снижение стоимости разработки моделей, созданных вручную. Мы рассматриваем методы поиска нейронной архитектуры, автоматического прореживания и квантования, которые могут превзойти ручное проектирование, требуя лишь минимальных усилий со стороны человека. Наконец, мы описываем универсальную методику, позволяющую эффективно поддерживать многие аппаратные платформы и соответствовать ограничениям эффективности без повторения дорогостоящих этапов поиска и переобучения.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

Baker Bowen, Gupta Otkrist, Naik Nikhil, Raskar Ramesh, 2016. Designing neural network architectures using reinforcement learning. arXiv preprint. arXiv: 1611.02167.

- Banner Ron, Nahshan Yury, Soudry Daniel*, 2019. Post training 4-bit quantization of convolutional networks for rapid-deployment. In: *Advances in Neural Information Processing Systems*, pp. 7950–7958.
- Bengio Yoshua, Léonard Nicholas, Courville Aaron*, 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint. arXiv:1308.3432*.
- Brock Andrew, Lim Theodore, Ritchie James M., Weston Nick*, 2017. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint. arXiv:1708.05344*.
- Bucilua Cristian, Caruana Rich, Niculescu-Mizil Alexandru*, 2006. Model compression. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541.
- Cai Han, Chen Tianyao, Zhang Weinan, Yu Yong, Wang Jun*, 2018a. Efficient architecture search by network transformation. In: *AAAI*.
- Cai Han, Yang Jiacheng, Zhang Weinan, Han Song, Yu Yong*, 2018b. Path-level network transformation for efficient architecture search. In: *ICML*.
- Cai Han, Gan Chuang, Wang Tianzhe, Zhang Zhekai, Han Song*, 2020a. Once for all: train one network and specialize it for efficient deployment. In: *International Conference on Learning Representations*.
- Cai Han, Zhu Ligeng, Proxyless N. A. S, Song Han*, 2019. Direct neural architecture search on target task and hardware. In: *International Conference on Learning Representations*.
- Cai Yaohui, Yao Zhewei, Dong Zhen, Gholami Amir, Mahoney Michael W., Zeroq Kurt Keutzer*, 2020b. A novel zero shot quantization framework. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13169–13178.
- Chen Guobin, Choi Wongun, Yu Xiang, Han Tony, Chandraker Manmohan*, 2017. Learning efficient object detection models with knowledge distillation. In: *Advances in Neural Information Processing Systems*, pp. 742–751.
- Chen Liang-Chieh, Collins Maxwell, Zhu Yukun, Papandreou George, Zoph Barret, Schroff Florian, Adam Hartwig, Shlens Jon*, 2018. Searching for efficient multi-scale architectures for dense image prediction. In: *Advances in Neural Information Processing Systems*, pp. 8699–8710.
- Chen Yu-Hsin, Krishna Tushar, Emer Joel S., Eyeriss Vivienne Sze*, 2016. An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits*.
- Chen Yu-Hsin, Yang Tien-Ju, Emer Joel, Sze Vivienne*, 2019a. Eyeriss v2: a flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9 (2), 292–308.
- Chen Yukang, Yang Tong, Zhang Xiangyu, Meng Gaofeng, Xiao Xinyu, Detnas Jian Sun*, 2019b. Backbone search for object detection. In: *Advances in Neural Information Processing Systems*, pp. 6642–6652.
- Cheong Robin, Daniel Robel*, 2019. Transformers. Zip: Compressing transformers with pruning and quantization. Technical report. Stanford University, Stanford, California.
- Courbariaux Matthieu, Bengio Yoshua*, 2016. Binarynet: training deep neural networks with weights and activations constrained to +1. *arXiv:1602.02830*.

- Courbariaux Matthieu, Bengio Yoshua, Binaryconnect Jean-Pierre David*, 2015. Training deep neural networks with binary weights during propagations. In: NIPS.
- Deng Jia, Dong Wei, Socher Richard, Li Li-Jia, Li Kai, Li Imagenet Fei-Fei*, 2009. A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE, pp. 248–255.
- Deng Lei, Li Guoqi, Han Song, Shi Luping, Xie Yuan*, 2020. Model compression and hardware acceleration for neural networks: a comprehensive survey. *Proceedings of the IEEE* 108 (4), 485–532.
- Denton Emily L., Zaremba Wojciech, Bruna Joan, LeCun Yann, Fergus Rob*, 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In: *Advances in Neural Information Processing Systems*, pp. 1269–1277.
- Elsken Thomas, Metzen Jan Hendrik, Hutter Frank*, 2018. Efficient multi-objective neural architecture search via Lamarckian evolution. *arXiv preprint. arXiv:1804.09081*.
- Engelbrecht Andries Petrus*, 2001. A new pruning heuristic based on variance analysis of sensitivity information. *IEEE Transactions on Neural Networks* 12 (6), 1386–1399.
- Frankle Jonathan, Carbin Michael*, 2018. The lottery ticket hypothesis: finding sparse, trainable neural networks. *arXiv preprint. arXiv:1803.03635*.
- Frankle Jonathan, Dziugaite Gintare Karolina, Roy Daniel, Carbin Michael*, 2020. Linear mode connectivity and the lottery ticket hypothesis. In: *International Conference on Machine Learning*. PMLR, pp. 3259–3269.
- Ghiasi Golnaz, Lin Tsung-Yi, Le Nas-fpn Quoc V.*, 2019. Learning scalable feature pyramid architecture for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045.
- Giles C. Lee, Omlin Christian W.*, 1994. Pruning recurrent neural networks for improved generalization performance. *IEEE Transactions on Neural Networks* 5 (5), 848–851.
- Girshick Ross*, 2015. Fast r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448.
- Golub Gene H., Van Loan Charles F.*, 1996. *Matrix Computations*. Johns Hopkins University Press, Baltimore and London.
- Gong Yunchao, Liu Liu, Yang Ming, Bourdev Lubomir*, 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint. arXiv:1412.6115*.
- Guo Yiwen, Yao Anbang, Chen Yurong*, 2016. Dynamic network surgery for efficient dnns. In: *Advances in Neural Information Processing Systems*, pp. 1379–1387.
- Guo Zichao, Zhang Xiangyu, Mu Haoyuan, Heng Wen, Liu Zechun, Wei Yichen, Sun Jian*, 2019. Single path one-shot neural architecture search with uniform sampling. *arXiv preprint. arXiv:1904.00420*.
- Han Song*, 2017. *Efficient methods and hardware for deep learning*.
- Han Song, Kang Junlong, Mao Huizi, Hu Yiming, Li Xin, Li Yubin, Xie Dongliang, Luo Hong, Yao Song, Wang Yu, et al.*, 2017. Ese: efficient speech recognition engine with sparse lstm on fpga. In: *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, pp. 75–84.
- Han Song, Liu Xingyu, Mao Huizi, Pu Jing, Pedram Ardavan, Horowitz Mark A., Dally William J.*, 2016. Eie: efficient inference engine on compressed deep neural

- network. In: Proceedings of the 43rd International Symposium on Computer Architecture. IEEE Press, pp. 243–254.
- Han Song, Mao Huizi, Dally William J.*, 2015a. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint. arXiv:1510.00149.
- Han Song, Pool Jeff, Tran John, Dally William*, 2015b. Learning both weights and connections for efficient neural network. In: Advances in Neural Information Processing Systems, pp. 1135–1143.
- Hassibi Babak, Stork David G.*, 1993. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. Morgan Kaufmann.
- He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian*, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
- He Yihui, Lin Ji, Liu Zhijian, Wang Hanrui, Li Li-Jia, Amc Song Han*, 2018. Automl for model compression and acceleration on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 784–800.
- He Yihui, Zhang Xiangyu, Sun Jian*, 2017. Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1389–1397.
- Hinton Geoffrey, Vinyals Oriol, Dean Jeff*, 2015. Distilling the knowledge in a neural network. arXiv preprint. arXiv:1503.02531.
- Howard Andrew, Sandler Mark, Chu Grace, Chen Liang-Chieh, Chen Bo, Tan Mingxing, Wang Weijun, Zhu Yukun, Pang Ruoming, Vasudevan Vijay, et al.*, 2019. Searching for mobilenetv3. In: ICCV 2019.
- Howard Andrew G., Zhu Menglong, Chen Bo, Kalenichenko Dmitry, Wang Weijun, Weyand Tobias, Andreetto Marco, Mobilenets Hartwig Adam*, 2017. Efficient convolutional neural networks for mobile vision applications. arXiv preprint. arXiv:1704.04861.
- Iandola Forrest N., Moskewicz Matthew W., Ashraf Khalid, Han Song, Dally William J., Squeezenet Kurt Keutzer*, 2016. Alexnet-level accuracy with 50x fewer parameters and < 1mb model size. arXiv preprint. arXiv:1602.07360.
- Ioffe Sergey, Szegedy Christian*, 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint. arXiv:1502.03167.
- Jacob Benoit, Kligys Skirmantas, Chen Bo, Zhu Menglong, Tang Matthew, Howard Andrew, Adam Hartwig, Kalenichenko Dmitry*, 2017. Quantization and training of neural networks for efficient integer-arithmetic-only inference. arXiv preprint. arXiv:1712.05877.
- Jaderberg Max, Vedaldi Andrea, Zisserman Andrew*, 2014. Speeding up convolutional neural networks with low rank expansions. arXiv preprint. arXiv:1405.3866.
- Judd Patrick, Albericio Jorge, Hetherington Tayler, Aamodt Tor M., Stripes Andreas Moshovos*, 2016. Bit-serial deep neural network computing. In: MICRO.
- Kim Yong-Deok, Park Eunhyeok, Yoo Sungjoo, Choi Taelim, Yang Lu, Shin Dongjun*, 2015. Compression of deep convolutional neural networks for fast and low power mobile applications. arXiv preprint. arXiv:1511.06530.
- Krizhevsky Alex, Hinton Geoffrey*, 2009. Learning multiple layers of features from tiny images.

- Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E., 2012. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.
- Lebedev Vadim, Ganin Yaroslav, Rakhuba Maksim, Oseledets Ivan, Lempitsky Victor, 2014. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint. arXiv:1412.6553.
- Lebedev Vadim, Lempitsky Victor, 2016. Fast convnets using group-wise brain damage. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2554–2564.
- LeCun Yann, Cortes Corinna, Burges Christopher JC, 2010. Mnist handwritten digit database. AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>.
- LeCun Yann, Denker John S., Solla Sara A., Howard Richard E., Jackel Lawrence D., 1989. Optimal brain damage. In: *NIPs*, vol. 2, pp. 598–605.
- Li Fengfu, Zhang Bo, Liu Bin, 2016a. Ternary weight networks. arXiv preprint. arXiv:1605.04711.
- Li Hao, Kadav Asim, Durdanovic Igor, Samet Hanan, Graf Hans Peter, 2016b. Pruning filters for efficient convnets. arXiv preprint. arXiv:1608.08710.
- Li Muyang, Lin Ji, Ding Yaoyao, Liu Zhijian, Zhu Jun-Yan, Han Song, 2020. Gan compression: efficient architectures for interactive conditional gans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5284–5294.
- Lillicrap Timothy P., Hunt Jonathan J., Pritzel Alexander, Heess Nicolas, Erez Tom, Tassa Yuval, Silver David, Wierstra Daan, 2015. Continuous control with deep reinforcement learning. arXiv preprint. arXiv:1509.02971.
- Lin Ji, Rao Yongming, Lu Jiwen, 2017. Runtime neural pruning. In: *NeurIPS*.
- Lin Zhouhan, Courbariaux Matthieu, Memisevic Roland, Bengio Yoshua, 2015. Neural networks with few multiplications. arXiv preprint. arXiv:1510.03009.
- Liu Chenxi, Chen Liang-Chieh, Schroff Florian, Adam Hartwig, Hua Wei, Yuille Alan L., Li Auto-deeplab Fei-Fei, 2019. Hierarchical neural architecture search for semantic image segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 82–92.
- Liu Chenxi, Zoph Barret, Neumann Maxim, Shlens Wei Hua Jonathon, Li Li-Jia, Fei-Fei Li, Yuille Alan, Huang Jonathan, Murphy Kevin, 2018. Progressive neural architecture search. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 19–34.
- Liu Chenxi, Zoph Barret, Shlens Wei Hua Jonathon, Li Li-Jia, Fei-Fei Li, Yuille Alan, Huang Jonathan, Murphy Kevin, 2017. Progressive neural architecture search. arXiv preprint. arXiv:1712.00559.
- Liu Hanxiao, Simonyan Karen, Vinyals Oriol, Fernando Chrisantha, Kavukcuoglu Koray, 2018a. Hierarchical representations for efficient architecture search. In: *ICLR*.
- Liu Hanxiao, Simonyan Karen, Darts Yiming Yang, 2018b. Differentiable architecture search. arXiv preprint. arXiv: 1806.09055.
- Liu Yifan, Chen Ke, Liu Chris, Qin Zengchang, Luo Zhenbo, Wang Jingdong, 2019a. Structured knowledge distillation for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2604–2613.

- Liu Zechun, Mu Haoyuan, Zhang Xiangyu, Guo Zichao, Yang Xin, Cheng Kwang-Ting, Metapruning Jian Sun*, 2019b. Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3296–3305.
- Ma Ningning, Zhang Xiangyu, Zheng Hai-Tao, Sun Jian*, 2018. Shufflenet v2: practical guidelines for efficient cnn architecture design. In: ECCV.
- Ma Xiaolong, Guo Wei Niu Fu-Ming, Lin Xue, Tang Jian, Ma Kaisheng, Ren Bin, Pconv Yanzhi Wang*, 2020. The missing but desirable sparsity in dnn weight pruning for real-time execution on mobile devices. In: AAAI, pp. 5117–5124.
- Mao Huizi, Han Song, Pool Jeff, Li Wenshuo, Liu Xingyu, Wang Yu, Dally William J.*, 2017. Exploring the granularity of sparsity in convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 13–20.
- Molchanov Pavlo, Tyree Stephen, Karras Tero, Aila Timo, Kautz Jan*, 2016. Pruning convolutional neural networks for resource efficient transfer learning. CoRR. arXiv:1611.06440 [abs].
- Molchanov Pavlo, Tyree Stephen, Karras Tero, Aila Timo, Kautz Jan*, 2017. Pruning convolutional neural networks for resource efficient transfer learning. In: International Conference on Learning Representations.
- Nagel Markus, van Baalen Mart, Blankevoort Tijmen, Welling Max*, 2019. Data-free quantization through weight equalization and bias correction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1325–1334.
- Niu Wei, Ma Xiaolong, Lin Sheng, Wang Shihao, Qian Xuehai, Lin Xue, Wang Yanzhi, Patdnn Bin Ren*, 2020. Achieving real-time dnn execution on mobile devices with pattern-based weight pruning. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 907–922.
- Pham Hieu, Guan Melody Y., Zoph Barret, Le Quoc V., Dean Jeff*, 2018. Efficient neural architecture search via parameter sharing. In: ICML.
- Ramachandran Prajit, Zoph Barret, Le Quoc V.*, 2017. Searching for activation functions. arXiv preprint. arXiv: 1710.05941.
- Rastegari Mohammad, Ordonez Vicente, Redmon Joseph, Xnor-net Ali Farhadi*, 2016. Imagenet classification using binary convolutional neural networks. In: European Conference on Computer Vision. Springer, pp. 525–542.
- Real Esteban, Aggarwal Alok, Huang Yanping, Le Quoc V.*, 2018. Regularized evolution for image classifier architecture search. arXiv preprint. arXiv:1802.01548.
- Romero Adriana, Ballas Nicolas, Ebrahimi Kahou Samira, Chassang Antoine, Gatta Carlo, Bengio Yoshua*, 2014. Fitnets: hints for thin deep nets. arXiv preprint. arXiv:1412.6550.
- Sandler Mark, Howard Andrew, Zhu Menglong, Zhmoginov Andrey, Chen Liang-Chieh*, 2018. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.
- Sanh Victor, Debut Lysandre, Chaumond Julien, Wolf Thomas*, 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

- Sharify Sayeh, Delmas Lascorz Alberto, Siu Kevin, Judd Patrick, Loom Andreas Mo-shovos*, 2018. Exploiting weight and activation precisions to accelerate convolutional neural networks. In: DAC.
- Sharma Hardik, Park Jongse, Suda Naveen, Lai Liangzhen, Chau Benson, Chandra Vikas, Esmailzadeh Hadi*, 2018. Bit fusion: bit-level dynamically composable architecture for accelerating deep neural networks. In: Proceedings of the 45th Annual International Symposium on Computer Architecture. IEEE Press, pp. 764–775.
- Simonyan Karen, Zisserman Andrew*, 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.
- Srinivas Suraj, Babu R. Venkatesh*, 2015. Data-free parameter pruning for deep neural networks. arXiv preprint. arXiv:1507.06149.
- Strubell Emma, Ganesh Ananya, McCallum Andrew*, 2019. Energy and policy considerations for deep learning in nlp. In: ACL.
- Szegedy Christian, Liu Wei, Jia Yangqing, Sermanet Pierre, Reed Scott, Anguelov Dragomir, Erhan Dumitru, Vanhoucke Vincent, Rabinovich Andrew*, 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
- Szegedy Christian, Vanhoucke Vincent, Ioffe Sergey, Shlens Jon, Wojna Zbigniew*, 2016. Rethinking the inception architecture for computer vision. In: CVPR.
- Tan Mingxing, Chen Bo, Pang Ruoming, Vasudevan Vijay, Le Mnasnet Quoc V.*, 2018. Platform-aware neural architecture search for mobile. arXiv preprint. arXiv:1807.11626.
- Tan Mingxing, Pang Ruoming, Le Efficientdet Quoc V.*, 2020a. Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790.
- Tan Zhanhong, Song Jiebo, Ma Xiaolong, Tan Sia-Huat, Chen Hongyang, Miao Yuan-qing, Wu Yifu, Ye Shaokai, Wang Yanzhi, Li Dehui, et al.*, 2020b. Pcnnet: pattern-based fine-grained regular pruning towards optimizing cnn accelerators. arXiv preprint. arXiv:2002.04997.
- Umuroglu Yaman, Rasnayake Lahiru, Bismo Magnus Sjalander*, 2018. A scalable bit-serial matrix multiplication overlay for reconfigurable computing. In: FPL.
- Vanhoucke Vincent, Senior Andrew, Mao Mark Z.*, 2011. Improving the Speed of Neural Networks on Cpus. In: Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop. In: Citeseer, vol. 1, p. 4.
- Wang Kuan, Liu Zhijian, Lin Yujun, Lin Ji, Haq Song Han*, 2018. Hardware-aware automated quantization. arXiv preprint. arXiv:1811.08886.
- Wen Wei, Wu Chunpeng, Wang Yandan, Chen Yiran, Li Hai*, 2016. Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems, pp. 2074–2082.
- Wu Bichen, Dai Xiaoliang, Zhang Peizhao, Wang Yanghan, Sun Fei, Wu Yiming, Tian Yuandong, Vajda Peter, Jia Yangqing, Fbnet Kurt Keutzer*, 2019. Hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10734–10742.
- Wu Jiaxiang, Leng Cong, Wang Yuhang, Hu Qinghao, Cheng Jian*, 2016. Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4820–4828.

- Xue Jian, Li Jinyu, Gong Yifan, 2013. Restructuring of deep neural network acoustic models with singular value decomposition. In: *Interspeech*, pp. 2365–2369.
- Yang Tien-Ju, Howard Andrew, Chen Bo, Zhang Xiao, Go Alec, Sze Vivienne, Netadapt Hartwig Adam, 2018. Platform-aware neural network adaptation for mobile applications. *arXiv preprint*. arXiv:1804.03230.
- Yu Jiahui, Autoslim Thomas Huang, 2019. Towards one-shot architecture search for channel numbers. *arXiv preprint*. arXiv:1903.11728.
- Yu Jiahui, Jin Pengchong, Liu Hanxiao, Bender Gabriel, Kindermans Pieter-Jan, Tan Mingxing, Huang Thomas, Song Xiaodan, Pang Ruoming le Quoc, 2020. Bignas: scaling up neural architecture search with big single-stage models. *arXiv preprint*. arXiv:2003.11142.
- Yu Jiecao, Lukefahr Andrew, Palframan David, Dasika Ganesh, Das Reetuparna, Scalpel Scott Mahlke, 2017. Customizing dnn pruning to the underlying hardware parallelism. *ACM SIGARCH Computer Architecture News* 45 (2), 548–560.
- Zagoruyko Sergey, Komodakis Nikos, 2016. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. *arXiv preprint*. arXiv:1612.03928.
- Zhang Chen, Li Peng, Sun Guangyu, Guan Yijin, Xiao Bingjun, Cong Jason, 2015. Optimizing fpga-based accelerator design for deep convolutional neural networks. In: *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 161–170.
- Zhang Shijin, Du Zidong, Zhang Lei, Lan Huiying, Liu Shaoli, Li Ling, Guo Qi, Chen Tianshi, Cambricon-x Yunji Chen, 2016a. An accelerator for sparse neural networks. In: *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, pp. 1–12.
- Zhang Xiangyu, Zou Jianhua, He Kaiming, Sun Jian, 2016b. Accelerating very deep convolutional networks for classification and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (10), 1943–1955.
- Zhang Xiangyu, Zhou Xinyu, Lin Mengxiao, Shufflenet Jian Sun, 2017. An extremely efficient convolutional neural network for mobile devices. *arXiv preprint*. arXiv:1707.01083.
- Zhong Zhao, Yan Junjie, Liu Cheng-Lin, 2017. Practical network blocks design with q-learning. *arXiv preprint*. arXiv:1708.05552.
- Zhou Shuchang, Wu Yuxin, Ni Zekun, Zhou Xinyu, Wen He, Dorefa-net Yuheng Zou, 2016. Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint*. arXiv:1606.06160.
- Zhu Chenzhuo, Han Song, Mao Huizi, Dally William J., 2016. Trained ternary quantization. *arXiv preprint*. arXiv: 1612.01064.
- Zoph Barret, Le Quoc V., 2016. Neural architecture search with reinforcement learning. *arXiv preprint*. arXiv:1611.01578.
- Zoph Barret, Vasudevan Vijay, Shlens Jonathon, Le Quoc V., 2017. Learning transferable architectures for scalable image recognition. *arXiv preprint*. arXiv:1707.07012.

Глава 5

Условная генерация изображений и управляемая генерация визуальных паттернов

Авторы главы:

Ган Хуа, Wormpex AI Research, Белвью, Вашингтон, США;
Донгдонг Чен, Microsoft Cloud & AI, Редмонд, Вашингтон, США

Краткое содержание главы:

- визуальный паттерн – это визуально различимая регулярность, которая повторяется предсказуемым образом;
- распознавание, обнаружение и синтез паттернов – три фундаментальные задачи в изучении зрительных образов;
- синтез паттернов является наиболее сложной задачей визуального моделирования;
- изучение разделенных представлений – ключ к более управляемому синтезу визуальных паттернов;
- изучение разделенных представлений без учителя требует соответствующих индуктивных предпосылок;
- более управляемый синтез паттернов приводит к более объяснимому их анализу.

5.1. ВВЕДЕНИЕ

Зрительное восприятие – сложная задача, поскольку естественные сцены состоят из огромного количества *визуальных паттернов* (visual pattern)¹, ко-

¹ Изначально словосочетание *visual pattern* в научно-технической литературе переводили как «визуальный образ», хотя в английском языке слово *pattern* имеет

торые часто сопровождают либо стохастические, либо детерминированные процессы, либо их комбинации. По определению визуальный паттерн – это *различимая* визуальная закономерность, чьи композиционные элементы в целом *повторяются* предсказуемым образом. В изучении визуальных паттернов есть три основные задачи: распознавание, обнаружение и синтез. Различение отличающихся визуальных паттернов соответствует задаче распознавания паттернов¹. Тот факт, что визуальные паттерны повторяются в окружающем мире, лежит в основе различных методов обнаружения паттернов. Синтез паттерна представляет собой задачу создания нового экземпляра визуального паттерна. Это влечет за собой процесс моделирования и описания глубинных процессов, которые определяют (и, следовательно, могут предсказывать) вариации визуального паттерна.

С точки зрения моделирования и обучения для выполнения трех вышеуказанных фундаментальных задач необходима разная минимальная информация. В частности, для распознавания паттернов нам достаточно идентифицировать наиболее отличительные визуальные признаки² паттерна, чтобы отличить его от других визуальных паттернов. Другими словами, успешное выполнение задач распознавания образов (например, классификации) не требует, чтобы представления признаков были исчерпывающими при описании каждой отдельной детали паттерна. Это одна из причин того, что в большинстве современных методов распознавания паттернов, если не во всех, применяется подход дискриминационного моделирования. Чтобы лучше это проиллюстрировать, мы предлагаем ознакомиться с несколькими примерами изображений на рис. 5.1. Очевидно, что, даже бегло взглянув на визуальные характеристики некоторых локальных участков изображения, мы уже можем достоверно распознать смысловую категорию этих изображений, даже если это стилизованные художественные произведения.

Моделирование паттернов требует немного больше усилий, поскольку задача заключается в обнаружении и локализации повторяющихся визуальных паттернов из набора изображений и видео без заранее определенного пространства гипотез (Yuan, 2011; Zhao et al., 2013). Задачам обнаружения паттернов присуще отсутствие обучающей разметки. Поэтому возникает потребность в моделях, способных установить относительно общие пред-

более широкий смысл – это одновременно и *узнаваемый* визуальный образ, и повторяющийся шаблон, и нечто среднее между ними. В последнее время это слово стали использовать почти как синоним слова «шаблон», особенно в технических дисциплинах, и часто применяют без перевода. Тем не менее нужно иметь в виду, что в зависимости от контекста слово «паттерн» может по-прежнему означать «образ». Строгое введение понятия «паттерн» в научный обиход выполнено в работе Ф. Т. Алескерова и др. «Анализ паттернов в статике и динамике. Часть 1: обзор литературы и уточнение понятия». <https://bijournal.hse.ru/data/2013/10/03/1277895965/1.pdf>. – Прим. перев.

¹ Здесь мы имеем в виду распознавание паттернов в узком смысле различения разных визуальных образов. Его также можно использовать в широком смысле, чтобы охватить все задачи анализа зрительных закономерностей.

² В самом деле, визуальный признак также можно рассматривать как атомарный визуальный паттерн.

ставления и метрическое пространство, в котором может быть проведена перцептивная группировка для выявления значимых семантических визуальных паттернов.



Рис. 5.1 ❖ Примеры, показывающие, что для распознавания изображения нам может понадобиться всего несколько отличительных визуальных признаков (также известных как *атомарные визуальные паттерны*). Даже глядя только на визуальные признаки в локальных областях изображений, мы хорошо распознаем семантические категории, к которым они относятся

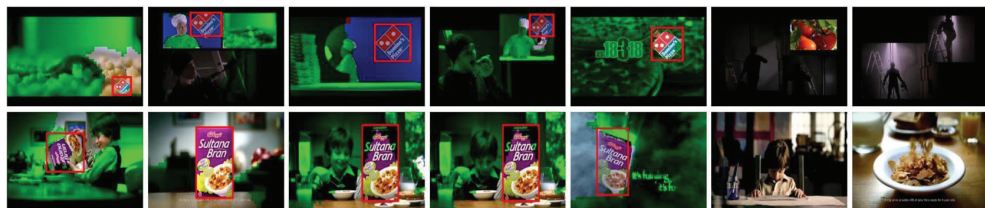


Рис. 5.2 ❖ Примеры обнаружения объектов в видеопоследовательности без разметки с использованием технологий, предложенных в (Zhao et al., 2013). Образцы изображений предоставлены авторами исследования

В некоторых работах также рассматривалась проблема обнаружения паттернов в задаче со слабой разметкой (Liu et al., 2010). Часто предоставляется только ограниченное количество образцов, иногда даже лишь на уровне кадра, указывающих, содержит ли кадр паттерн, без указания, где последний расположен. Помимо моделирования характеристик визуальных паттернов, было доказано, что пространственная, временная и/или пространственно-временная контекстуальная информация способствует решению задач обнаружения визуальных паттернов. Эффективная модель для обнаружения визуального паттерна требует представления признаков, которое ищет компромисс между всесторонним описанием паттерна и достаточной дискриминацией, чтобы паттерны можно было эффективно сгруппировать и отличить от фона. Следовательно, существующие подходы к обнаружению паттернов могут быть либо генеративными (Zhao et al., 2013), либо дискриминационными (Weng et al., 2018), либо их комбинацией.

Синтез паттернов требует всестороннего моделирования целевых визуальных паттернов, поскольку в конечном итоге модель должна фиксировать каждую деталь вместе с вариациями визуальных паттернов. В подходе к проблеме синтеза паттернов, основанном на обучении, часто используют генеративное моделирование. Хотя в генеративном моделировании были

достигнуты огромные успехи либо с использованием традиционных статистических методов (Zhu et al., 1998; Van de Wouwer et al., 1999; Zhu, 2003; Guo et al., 2003), либо с использованием более поздних методов глубокого обучения (Kingma and Welling, 2014; Goodfellow et al., 2014), процессы генерации (или выборки из моделей) часто определяются случайным процессом, когда целенаправленно генерировать визуальные паттерны сложно, если вообще возможно.

В этой главе мы сосредоточим наше обсуждение на том, как мы можем добиться более управляемого синтеза визуальных паттернов с помощью генерации условного изображения на основе глубокого обучения. Здесь «управляемый» означает, что существует способ, которым мы можем намеренно задать значение определенного параметра или подмножества параметров, чтобы управлять генерацией образцов визуальных паттернов по определенным семантическим, физическим и/или геометрическим измерениям (также известным как *факторы*) – таким как выражение лица, цвет и позы. Мы достигаем такой управляемости за счет *глубокой условной генерации изображений* (deep conditional image generation) со структурой кодер–декодер, где для управления генерацией выборок целевых визуальных паттернов определяется вероятностное пространство над разделенным векторным представлением. Обучение такого *разделенного представления* (disentangled representation) является сложной задачей, которая остается предметом активных исследований. Хотя во многих ранее опубликованных работах авторы утверждали, что научились разделять векторное представление в процессе обучения без учителя, Локателло и др. (Locatello et al., 2019) показывают, что изучение разделенного представления без учителя теоретически невозможно без *индуктивных предпосылок* (inductive bias) как в моделях, так и в наборах данных. Мы посвятили наше исследование тому, как в реальных приложениях компьютерного зрения подобные индуктивные предпосылки могут быть введены путем обучения с учителем, частичного обучения и самообучения, чтобы изучить разделенное представление для управляемого синтеза визуальных паттернов. Применения, которые мы рассматриваем, включают перенос стиля изображения/видео, преобразование текста в изображение и синтез лица. Тем не менее мы надеемся, что выводы, полученные в результате этих исследований, помогут решить и другие задачи в различных приложениях.

Оставшаяся часть главы будет организована следующим образом: в разделе 5.2 мы представляем краткий исторический обзор обучения моделей визуальных паттернов. Затем в разделах 5.3 и 5.4 раскрываются основы традиционных генеративных моделей, основанных на статистическом обучении, и глубоких генеративных моделей. Далее, в разделе 5.5, мы расскажем, как использовать глубокие генеративные модели для обучения и синтеза визуальных паттернов в рамках условной генерации изображений. В разделе 5.6 мы опишем три конкретных примера с разными уровнями вовлеченности учителя в обучение, чтобы показать, как можно ввести индуктивную предпосылку для изучения разделенных представлений и последующего управляемого синтеза визуальных паттернов. Наконец, мы делаем выводы и предположения о направлении будущих исследований в разделе 5.7.

5.2. ИЗУЧЕНИЕ ВИЗУАЛЬНЫХ ПАТТЕРНОВ: КРАТКИЙ ИСТОРИЧЕСКИЙ ОБЗОР

Ранние исследования зрительных паттернов уходят корнями в прошлое на несколько десятилетий. Наиболее распространенным подходом было использование байесовских структур и разработка множества явных моделей для моделирования визуальных паттернов. Отдельно можно отметить пионерские работы (Grenander, 1976; Cooper, 1979; Fu, 1982), в которых для моделирования визуальных паттернов использовались статистические модели. В конце 1980-х и начале 1990-х гг. становятся популярными модели изображений. Первые такие модели предполагали локальную и кусочную гладкость естественных изображений и были описаны во многих исследовательских работах. Например, в работах (Blake, Zisserman, 1987; Terzopoulos, 1983) предлагаются физически обоснованные модели, а в (Poggio et al., 1985) используется теория регуляризации. В работе (Mumford, Shah, 1989) эта задача сформулирована как энергетическая функция и решена методом минимизации энергии.

Затем благодаря двум влиятельным исследованиям вышеупомянутые концепции начали сближаться со статистическими описательными моделями. Одним из этих исследований является моделирование *марковских случайных полей* (Markov random fields, MRF) (Besag, 1974; Cross, Jain, 1983). Авторы этих работ исходили из того, что шаблон текстуры следует стохастическому, возможно периодическому, двумерному полю изображения, и исследовали марковские случайные поля в качестве моделей текстур. *Модель текстуры* определяется как математическая процедура, способная создать и описать текстурное изображение. Также получила известность работа (Geman, Geman, 1984), в которой проводится аналогия между изображениями и системами статистической механики и формулируется моделирование паттернов как задача распределения и выборки Гиббса в рамках байесовской структуры. Более конкретно, значения пикселей, а также наличие и ориентация краев рассматриваются как состояния атомов или молекул в физической системе, подобной кристаллической решетке, а назначение энергетической функции в физической системе определяет распределение Гиббса. Поскольку распределение Гиббса эквивалентно MRF, его также можно рассматривать как модель изображения MRF. Однако у этих работ есть два ограничения: 1) марковские модели случайных полей основаны на парных кликах, поэтому они не могут очень хорошо характеризовать естественные изображения; 2) выборка Гиббса занимает очень много времени, что затрудняет ее применение в реальных системах. Существуют и другие вероятностные модели, предложенные для обучения представлениям визуальных паттернов, такие как деформируемые шаблоны для лица (Yuille, 1991), глаз (Xie et al., 1994) и объектов (Shackleton, 1994). По сравнению с гомогенными (однородными) моделями MRF деформируемые шаблоны неоднородны.

В стремлении справиться с вычислительной сложностью вышеупомянутых описательных моделей были предложены генеративные модели, которые

постулируют скрытые переменные как причины сложных зависимостей в не-обработанных сигналах. Простая иллюстрация этого подхода изображена на рис. 5.3. Возьмем, к примеру, человека – у него обычно бывает одна голова (обозначим ее через h), одно туловище (b), две руки и две ноги (a_l, a_r, l_l, l_r). Описательная модель рассматривает совместное распределение пяти частей $p(h, b, a_l, a_r, l_l, l_r)$ без понимания скрытого понятия «человек». Напротив, генеративные модели считают пять частей условно зависимыми от скрытой переменной *human*, обозначающей человека, а затем формализуют их с помощью модели условной вероятности $p(h, b, a_l, a_r, l_l, l_r | d)$.

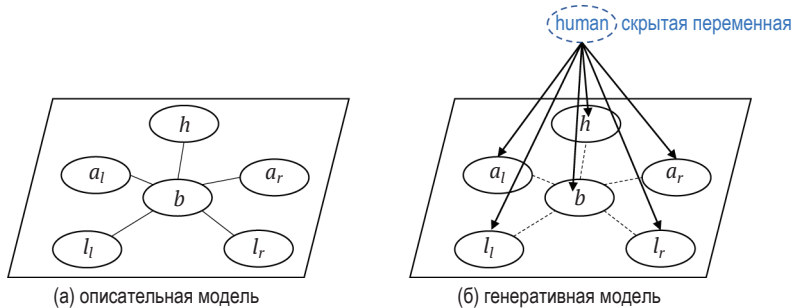


Рис. 5.3 ❖ Простая иллюстрация различий между описательными моделями и генеративными моделями с точки зрения представлений. В отличие от описательных моделей, генеративные модели вводят скрытые переменные для моделирования сильной зависимости в наблюдаемых изображениях

К типичным представителям генеративных моделей относятся разреженное кодирование (Roweis, Ghahramani, 1999; Hoyer, Hyvärinen, 2002; Manat, Zhang, 1993), вейвлетное представление изображений (Do, Vetterli, 2003; Lu et al., 1992), анализ главных компонент (principle component analysis, PCA) (Kambhatla, Leen, 1997; Kong et al., 2005), анализ независимых компонент (independent component analysis, ICA) (Hyvärinen, 1999; Hyvärinen, Oja, 2000) и случайная модель колледжа (Lee et al., 2001). Такие модели предполагают, что изображение может быть представлено серией базовых элементов. Таким образом, размер представления может быть значительно уменьшен за счет проецирования исходного пространства изображения в скрытое пространство. Следовательно, снижается стоимость вычислений. Во многих статьях генеративные модели часто неотделимы от описательных моделей, поскольку скрытые переменные нередко характеризуются описательной моделью. Например, схема разреженного кодирования представляет собой двухуровневую генеративную модель и предполагает, что базовые элементы изображений представлены независимыми и одинаково распределенными скрытыми переменными. В скрытых марковских моделях, предназначенных для моделирования речи и движения, роль скрытого слоя играет цепь Маркова.

В последнее время благодаря использованию наборов больших данных и быстродействующего вычислительного оборудования глубокое обучение достигло больших успехов во многих задачах искусственного интеллекта,

таких как визуальное распознавание (Szegedy et al., 2015; He et al., 2016) и обнаружение объектов (Girshick, 2015; Ren et al., 2015). По той же причине было предложено множество генеративных моделей на основе глубоких нейронных сетей, включая пиксельную CNN (Van den Oord et al., 2016), вариационный автокодировщик (VAE) (Doersch, 2016) и генеративно-состязательные сети (Goodfellow et al., 2014). По сути, чтобы генерировать высококачественные изображения, такие глубокие модели должны научиться запоминать в своем пространстве весов визуальные паттерны и лежащие в их основе структуры. В какой-то степени лучшее качество генерации эквивалентно лучшему обучению паттернам. В следующих разделах мы кратко представим основы классических и глубоких генеративных моделей.

5.3. КЛАССИЧЕСКИЕ ГЕНЕРАТИВНЫЕ МОДЕЛИ

С математической точки зрения (Hua, 2020) классические генеративные модели – это статистические модели, нацеленные на моделирование совместного распределения вероятностей $p(X, \mathbf{z}|\theta)$, где X – наблюдаемая многомерная переменная, \mathbf{z} – вышеупомянутая скрытая переменная, а θ – параметры модели, подлежащие оптимизации. Скрытая переменная Z может представлять различные *значимые факторы* (cofounder). Например, если Z – это метки классов, генеративные модели также можно использовать для задач классификации. Но цели моделирования генеративных моделей отличаются от дискриминационных моделей, которые напрямую моделируют условное распределение $p(Z|X, \theta)$.

Как и в случае с любыми статистическими моделями, *обучение и вывод* являются двумя фундаментальными проблемами, которые необходимо решать при разработке и использовании генеративных моделей. Обучение представляет собой процесс определения параметров этих генеративных моделей на основе данных. Когда нам доступны *полные данные*, т. е. когда X и Z наблюдаются как пара выборок, подбор параметров часто формализуется как стандартная задача поиска максимального правдоподобия. На практике чаще встречаются неполные данные, когда наблюдается только X , а целевая переменная Z не видна в выборке данных. Это задача поиска максимального правдоподобия с неполными данными. Такая проблема нередко решается с помощью оригинального алгоритма *максимизации ожидания* (expectation maximization, EM) (Dempster et al., 1977), согласно которому E -шаг и M -шаг итеративно выполняются для максимизации правдоподобия неполных данных. С точки зрения оптимизации такой итеративный процесс можно рассматривать как суррогатный процесс оптимизации.

E -шаг, согласно наименованию, вычисляет ожидание вероятности данных по распределению скрытых или целевых переменных. Это делается путем первого выполнения шага *вывода*, т. е. вычисления апостериорной вероятности $p(Z|X, \theta_c)$ с учетом текущего значения параметра θ_c . Тогда мы имеем:

$$E(\theta|\theta_c) = \int_{\mathbf{z}} p(X, \mathbf{z}|\theta) p(\mathbf{z}|X, \theta_c) d\mathbf{z}. \quad (5.1)$$

Затем M -шаг максимизирует $E(\theta|\theta_c)$ для получения обновленных параметров:

$$\theta_c^{new} = \operatorname{argmax}_{\theta} E(\theta|\theta_c). \quad (5.2)$$

Эти два шага повторяются до сходимости. Данная итерация представляет собой суррогатный процесс максимизации $\mathcal{L}(X|\theta)$, который монотонно не убывает. Поскольку он, очевидно, ограничен сверху, процесс гарантированно сходится. Существует байесовский ЕМ-алгоритм на основе вероятностного вариационного анализа (Ghahramani and Beal, 2001). Действительно, M -шаг не обязательно должен полностью решать задачу максимизации. Вместо этого ему нужно только найти новый параметр θ_c^{new} , который имеет большее значение $E(\theta_c^{new}|\theta_c)$, чем $E(\theta_c|\theta_c)$. Это так называемый *обобщенный ЕМ-алгоритм*.

Вывод апостериорной вероятности $p(Z|X, \theta_c)$ будет иметь решение в замкнутой форме в ограниченных случаях. Например, когда *приор*¹ (prior) $p(Z)$ и распределение правдоподобия $p(X|Z)$ сопряжены, то апостериорное распределение будет иметь ту же форму, что и априорное. Такие сопряженные априорные значения встречаются, когда распределения ограничены сопряженным экспоненциальным семейством (Ghahramani, Beal, 2001). Однако для более общих распределений часто бывает трудно вычислить апостериорную зависимость в замкнутой форме.

Когда вывод в замкнутой форме невозможен, часто можно прибегнуть к численному решению, например используя методы Монте-Карло с цепями Маркова (МСМС), наподобие выборки Гиббса, для получения выборок из этого распределения, а затем найти интеграл в уравнении (5.1) численно. Хинтон (Hinton, 2002) представил иной метод, а именно контрастную дивергенцию, чтобы использовать одноступенчатую выборку вместо полной выборки МСМС. Это могло бы значительно ускорить процесс обучения с определенной гарантией сходимости.

5.4. ГЛУБОКИЕ ГЕНЕРАТИВНЫЕ МОДЕЛИ

Глубокие генеративные модели также нацелены на моделирование распределения X со скрытыми переменными Z . Принципиальное отличие от вышеупомянутых традиционных генеративных методов заключается в том, что вместо созданных фактически вручную моделей на основе вейвлетов и разреженного кодирования используются глубокие нейронные сети, чья способность к обучению намного выше.

¹ Под термином *приор* понимается пространство гипотез (или если речь идет о вероятностях, то пространство вероятностей), в котором осуществляется поиск, и то, какие гипотезы из этого пространства мы предпочитаем в большей или меньшей степени, или каким исходам *априорно* присваиваем ту или иную вероятность. Например, если у нас есть исходное изображение x_{orig} и мы ищем ближайший элемент в сгенерированном нейросетью пространстве X , то в зарубежной литературе X будет называться *deer image prior*. – Прим. перев.

Автокодировщик

Мы начинаем знакомство с предметом с простых автокодировщиков (Wang et al., 2016). Строго говоря, они не являются типичными генеративными моделями, но это облегчит понимание других глубоких генеративных моделей. Как показано на рис. 5.4, *автокодировщик* обычно состоит из двух частей: *кодировщика* (энкодера, encoder) и *декодера* (decoder). Оба они состоят из уложенных в стек полносвязных или сверточных слоев. Кодировщик постоянно уменьшает размерность до меньшего скрытого представления z (код), а затем декодер симметрично восстанавливает входное изображение из скрытого представления в исходное разрешение. Необходимо подчеркнуть, что цель автоэнкодера состоит не просто в том, чтобы восстановить исходные изображения с помощью тривиальной функции идентификации, а в том, чтобы изучить лежащие в их основе визуальные паттерны, чтобы мы могли генерировать некоторые новые изображения из изученного скрытого пространства. Для достижения этой цели было предложено множество вариантов автокодировщика. Первый и наиболее типичный – это сжимающий (или неполный) автокодировщик (Zhai et al., 2018), который требует, чтобы размер скрытого кода был значительно меньше размера входных данных. Следовательно, скрытый код должен быть достаточно информативным для представления входного изображения, иначе потери при восстановлении будут очень большими. Второй популярный вариант – автокодировщик с шумоподавлением (Vincent et al., 2008), когда в обучающий набор данных преднамеренно добавляют шумы, чтобы автокодировщик научился шумоподавлению. Вместо изменения обучающего набора данных третий вариант – разреженный автокодировщик (Ng et al., 2011) – учитывает уровень разреженности при вычислении потерь и поощряет минимизацию количества активных единиц в кодовом слое, изучая таким образом разреженное представление набора данных.

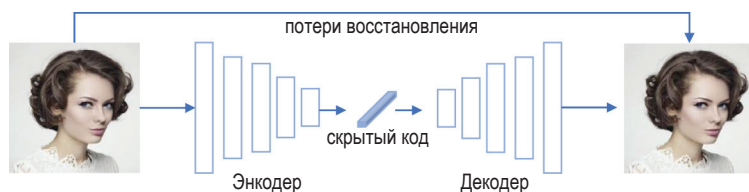


Рис. 5.4 ❖ Простая иллюстрация типичной глубокой генеративной модели автокодировщика, которая сначала кодирует входное изображение в скрытый код, а затем восстанавливает исходное изображение из скрытого кода с помощью декодера

Вариационный автокодировщик

Хотя вышеупомянутые автокодировщики могут успешно отображать обучающее изображение в пространство представлений, не существует явной вероятностной модели, связанной с этим пространством. Следовательно, случайная выборка из этого пространства представлений не может гарантировать

осмысленность выбора из пространства изображения при подаче выборочного кода в декодер. Чтобы решить эту проблему, *вариационный автокодировщик* (variational auto-encoder, VAE) (Doersch, 2016) заставляет изученное скрытое распределение следовать распределению Гаусса $P(z)$. Тогда цель VAE – максимизировать вероятность (максимальное правдоподобие) каждого X в обучающем наборе в рамках всего генеративного процесса:

$$P(X) = \int P(X|z; \theta)P(z)dz. \quad (5.3)$$

Решение интеграла (5.3) по z является нетривиальной задачей. На практике для большинства z значение $P(X|z)$ будет близко к нулю и, следовательно, почти не будет влиять на оценку $P(x)$. Таким образом, ключевая идея VAE состоит в том, чтобы попытаться выбрать значения z , которые, вероятно, произвели X , и вычислить $P(x)$ только для них. С этой целью вводится новая функция $Q(z|X)$, которая принимает значение X и формирует распределение по значениям z , которые, вероятно, будут генерировать X . Учитывая, что пространство значений z при Q может быть намного меньше, чем при предшествующем $P(z)$, вычисление $E_{z \sim Q}P(X|z)$ происходит относительно проще.

В основе вариационных байесовских методов лежит изучение дивергенции Кульбака–Лейблера между $P(z|X)$ и $Q(z)$, чтобы получить окончательную цель оптимизации.

$$\mathcal{D}[Q(z) \parallel P(z|X)] = E_{z \sim Q}[\log Q(z) - \log P(z|X)]. \quad (5.4)$$

Применяя правило Байеса к $P(z|X)$, в это уравнение можно включить и $P(X)$, и $P(X|z)$:

$$\mathcal{D}[Q(z) \parallel P(z|X)] = E_{z \sim Q}[\log Q(z) - \log P(X|z) - \log P(z)] + \log P(X). \quad (5.5)$$

Так как $\log P(X)$ не зависит от z , его можно вынести из ожидания. Перегруппировав члены с обеих сторон уравнения, мы получаем:

$$\log P(X) - \mathcal{D}[Q(z) \parallel P(z|X)] = E_{z \sim Q}[\log P(X|z)] - \mathcal{D}[Q(z) \parallel P(z)]. \quad (5.6)$$

Фактически Q может быть любым распределением, а не только распределением, которое хорошо отображает X в представления z , способные воспроизвести X . Поскольку конечной целью вывода является $P(X)$, имеет смысл построить Q , которое зависит от X и делает $\mathcal{D}[Q(z) \parallel P(z|X)]$ малым:

$$\log P(X) - \mathcal{D}[Q(z|X) \parallel P(z|X)] = E_{z \sim Q}[\log P(X|z)] - \mathcal{D}[Q(z|X) \parallel P(z)]. \quad (5.7)$$

Уравнение (5.7) является основой VAE. В частности, левая часть состоит из целевого логарифма $\log P(X)$, который мы хотим оптимизировать, плюс член ошибки, который заставляет Q создавать z , способные воспроизвести заданное X . Этот член ошибки становится небольшим, если Q моделируется высокоэффективной глубокой сетью. А правая часть – это предмет оптимизации с помощью стохастического градиентного спуска при правильном выборе Q . Фактически правая часть уравнения (5.7) принимает форму, похожую на автокодировщик, т. е. Q «кодирует» X в z , а P «декодирует» X из z .

Чтобы оптимизировать правую часть при помощи стохастического градиентного спуска, обычную форму $Q(z|X)$ представляют в виде $Q(z|X) = \mathcal{N}(z|\mu(X; \vartheta), \Sigma(X; \vartheta))$, где μ и Σ – произвольные детерминированные функции с параметрами ϑ , которые можно извлечь из данных. На практике μ и Σ снова реализуются через нейронные сети, и Σ ограничивается диагональной матрицей. Градиент первого члена в правой части уравнения оценивается путем выборки различных значений X из набора данных D и одного важного «трюка перепараметризации», как изображено на рис. 5.5.

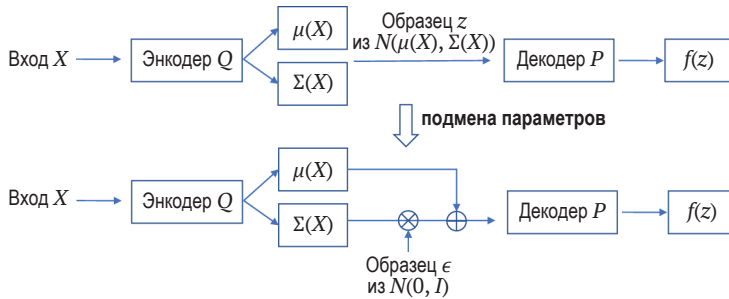


Рис. 5.5 ❖ «Трюк перепараметризации», используемый в вариационном автокодировщике для аппроксимации стохастического градиента во время обучения

Во время тестирования, если нужно сгенерировать новые образцы изображения, VAE случайным образом выбирает представления $z \sim \mathcal{N}(0, I)$ и вводит их в декодер. То есть в данном случае кодировщик со всеми его операциями умножения и сложения полностью отсутствует. Для получения более подробной информации мы рекомендуем читателям ознакомиться с работами Дорша (Doersch, 2016).

Генеративно-сопоставительная сеть

В последние годы широкую популярность приобрели *генеративно-сопоставительные сети* (generative adversarial networks, GAN), предложенные в новаторской работе Гудфеллоу и др. (Goodfellow et al., 2014). По сравнению с предыдущими генеративными моделями, здесь ключевая идея состоит в том, чтобы ввести вспомогательную сеть дискриминатора, помогающую в обучении целевой генеративной сети. Другой ключевой идеей является стратегия обучения противника, как показано на рис. 5.6, т. е. сеть дискриминатора учится различать, взята ли выборка из реального распределения данных или создана генерирующей сетью, в то время как генерирующая сеть пытается генерировать все более реалистичные изображения, чтобы обмануть сеть дискриминатора. Таким образом, эти две сети – генератор и дискриминатор – состязаются друг с другом в игре с нулевой суммой.

Математически это можно рассматривать как задачу минимаксной оптимизации с функцией ценности $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \quad (5.8)$$

где $D(x)$ представляет собой вероятность того, что x исходит из реального распределения данных, а не из сгенерированного $G(z)$. Подобно VAE, $p_z(z)$ представляет собой prior, из которого следует производить выборку и который может быть задан как нормальное распределение. Во время обучения и вывода случайно выбранное представление z передают в G для создания изображения. На практике G и D обучают альтернативным способом, т. е. сперва обучают G при неизменном D , затем обучают D при неизменном G . После обучения сеть дискриминатора отключают, и для создания новых изображений применяется только G .

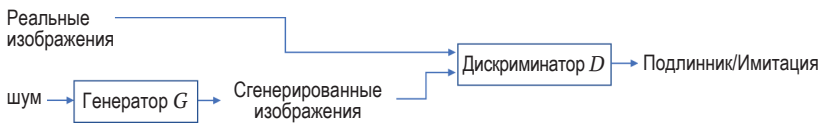


Рис. 5.6 ❖ Состязательное обучение генеративно-состязательных сетей: сеть-дискриминатор пытается научиться отличать реальное входное изображение от сгенерированного сетью-генератором, в то время как сеть-генератор пытается создавать более реалистичные изображения, чтобы обмануть сеть-дискриминатор

По сути, в процессе обучения G пытается достичь реального распределения данных, как показано на рис. 5.7. В идеальном случае, когда G может точно имитировать реальное распределение данных, D не в состоянии отличить сгенерированное изображение от реального, т. е. классифицирует изображение как настоящее/поддельное с вероятностью случайного угадывания 0,5.

Несмотря на громкий успех, обучение хорошей модели GAN оказывается не таким уж простым делом. За последнее время было предложено много вариантов улучшения качества и надежности GAN (Mirza, Osindero, 2014; Chen et al., 2016; Metz et al., 2016; Arjovsky et al., 2017; Bao et al., 2017). Более полное обобщение можно найти в (Wang et al., 2019).

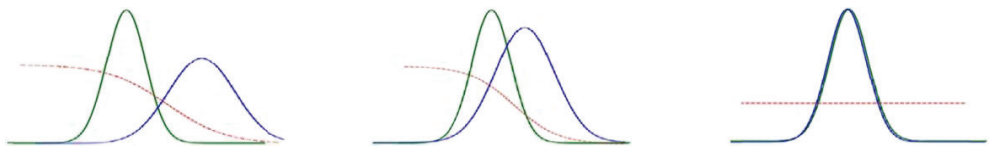


Рис. 5.7 ❖ Идеальная эволюция обучения генеративно-состязательных сетей. Зеленая линия – реальное распределение данных, синяя линия – распределение данных генеративной сети, а красная пунктирная линия – граница разделения выводов дискриминатора. В идеальном случае (последний столбец), когда генерирующая сеть научилась полностью имитировать реальное распределение, сеть дискриминатора не может различить реальное и сгенерированное распределения данных

5.5. ГЛУБОКАЯ УСЛОВНАЯ ГЕНЕРАЦИЯ ИЗОБРАЖЕНИЙ

Генеративные модели, о которых было сказано в предыдущих разделах, предназначены для безусловной генерации изображений, т. е. новые изображения генерируются на основе случайных кодированных представлений, выбранных из приора без каких-либо условий. Другими словами, происходит неуправляемый синтез визуальных паттернов. Напротив, в этой главе мы сосредоточимся на том, как добиться управляемого синтеза визуальных паттернов путем условной генерации изображений.

В отличие от безусловной генерации изображений, условная генерация (Isola et al., 2017; Zhu et al., 2017; Chen et al., 2017, 2020) накладывает на процесс генерации дополнительные входные условия. Эта методика охватывает широкий спектр проблем компьютерного зрения, таких как переход от контуров к изображению (Isola et al., 2017), перенос стиля (Gatys et al., 2015; Chen et al., 2018), раскрашивание изображений (He et al., 2018), управляемая обработка изображений (Fan et al., 2018, 2019; Chen et al., 2020), восстановление изображений (Wan et al., 2020), семантический синтез (Tan et al., 2020) и синтез и редактирование лиц (Tan et al., 2020).

Как показано на рис. 5.8, при заданных входных условиях, таких как текстовое описание, векторы атрибутов и изображения, условные генеративные модели нацелены на создание выходного изображения, удовлетворяющего условиям. Но, как и в случае безусловной генерации изображений, изучение и моделирование внутренних визуальных паттернов по-прежнему является важным фактором, определяющим качество генерации. Благодаря более эффективному моделированию визуальных паттернов генерация условного изображения, по сути, представляет собой процесс выборки и рекомпозиции ограниченного паттерна.

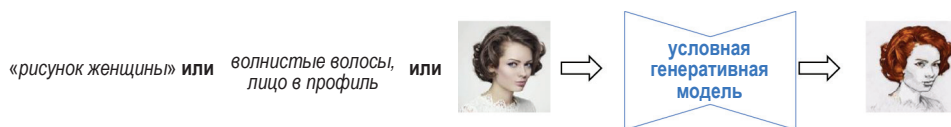


Рис. 5.8 ❖ Иллюстрация типичной схемы генерации условного изображения. Исходя из входных условий, таких как текстовое описание, атрибуты или изображения, условная генеративная модель должна генерировать выходное изображение, соответствующее условным требованиям

Тем не менее композиция естественных изображений включает в себя множество визуальных искажений, таких как положение (поза), освещение и форма. Чтобы побудить генеративную модель лучше изучить лежащий в основе паттерн, применяется разделение (иногда говорят «распутывание») в скрытом пространстве встраивания, которое является ключевым принципом проектирования и широко используется во многих существующих методах генерации условных изображений (Yan et al., 2016; Chen et al., 2017b;

Бао и др., 2018; Ма и др., 2018). Но реализация разделения – нетривиальная задача. Фактически показано (Locatello et al., 2019), что теоретически невозможно добиться обучаемого без учителя разделения без индуктивных предпосылок как в моделях, так и в наборах данных. Поэтому часто приходится разрабатывать специальную структуру сети и рецепт обучения с учителем, частичного привлечения учителя или самообучения.

5.6. РАЗДЕЛЕННЫЕ ПРЕДСТАВЛЕНИЯ В УПРАВЛЯЕМОМ СИНТЕЗЕ ПАТТЕРНОВ

Далее мы хотим представить три примера того, как можно ввести индуктивную предпосылку в изучение разделенных представлений при глубокой условной генерации изображений, применяемой в управляемом синтезе визуальных паттернов. Исследуемые нами области применения включают перенос стиля (раздел 5.6.1), создание изображений по описанию (раздел 5.6.2) и синтез лица с сохранением привязки к личности (раздел 5.6.3).

5.6.1. Разделение визуального содержания и стиля

Перенос стиля (Gatys et al., 2015; Johnson et al., 2016; Chen et al., 2020) – типичная задача создания условного изображения. Как показано на рис. 5.9, речь идет о переносе стиля изображения-источника на изображение-приемник с другим контентом при сохранении исходной семантической структуры приемника. Основные задачи, которые приходится решать при переносе стиля, заключаются в моделировании визуальных паттернов изображения-источника стиля и разделении содержимого и стиля изображения-приемника. Перенос стиля как таковой сводится к повторной выборке изученного паттерна стиля в соответствии с ограничением структуры контента.



Рис. 5.9 ❖ Иллюстрация переноса стиля. Перенос представляет собой визуализацию изображения-приемника в соответствии со стилем изображения-источника при сохранении исходной структуры приемника. Источник: Chen et al., 2020

После новаторской работы Леона Гэтиса (Gatys et al., 2015) перенос стиля с использованием сверточных нейронных сетей вызвал волну интереса как в академических кругах, так и в промышленности. В своем исследовании Гэтис и его коллеги оригинально применили предварительно обученную сверточную нейросеть для разложения изображения на компоненты *контента* и *стиля*. В частности, они рассматривают корреляционные матрицы Грамма откликов признаков в разных слоях как иерархическое представление стиля. Путем последующего сопоставления структуры контента одного изображения с его ответом на высокоуровневый признак они моделируют перенос стиля как проблему оптимизации, т. е. поиск сгенерированного изображения, которое имеет такие же грам-матрицы признаков, как источник стиля, и аналогичные высокоуровневые свойства содержимого, как изображение-приемник. Этот метод, основанный на оптимизации, может дать очень впечатляющие результаты стилизации и намного лучше, чем традиционные методы. Однако генерация занимает очень много времени из-за процесса оптимизации, что накладывает большие ограничения на реальные приложения.

Для ускорения процесса генерации были предложены различные методы, основанные на сетях прямого распространения и аппроксимирующие описанную выше процедуру оптимизации (Johnson et al., 2016; Ulyanov et al., 2016). При таком подходе результаты стилизации могут быть получены путем непосредственной подачи контента изображения в сеть прямого распространения; следовательно, это намного быстрее, чем в методах, основанных на оптимизации. Однако такие сети с прямым распространением обучаются по принципу «черного ящика», а компоненты контента и стиля (визуальные паттерны) в обученных сетях сильно связаны. Это не только не позволяет нам изучить явное представление стиля или контента, но и делает такие сети способными одновременно фиксировать только определенный стиль.

Исходя из этих соображений, Чен и др. (Chen et al., 2017) разработали новую разделенную сетевую структуру для изучения явного представления каждого стиля управляемым образом, что, естественно, поддерживает перенос нескольких стилей. Это соответствует концепции *текстона* в классическом синтезе текстуры, и мы предлагаем использовать последовательности банков фильтров для представления изображений разных стилей. Все каналы в одном банке фильтров можно рассматривать как основы стиливых элементов, таких как узор текстуры и штрихи, в одном источнике стиля. Затем выполняется процесс стилизации путем свертки соответствующих банков фильтров с картами признаков контента, что аналогично операции свертки между текстонам и дельта-функцией в пространстве изображения для синтеза текстуры (как показано на рис. 5.11).

Разделение стиля и контента

Фреймворк разделения на ветви детально изображен на рис. 5.10. По сути, он состоит из трех частей: одного общего кодировщика \mathcal{E} , слоя банка стилей \mathcal{K} и одного общего декодера \mathcal{D} . Чтобы заставить сеть явным образом разделить контент и стиль, мы создаем две обучающие ветви – реконструкции $\mathcal{E} \rightarrow \mathcal{D}$ и стилизации $\mathcal{E} \rightarrow \mathcal{K} \rightarrow \mathcal{D}$. Входное изображение I , которое является

источником контента, сначала преобразуется в пространство признаков F с использованием подсети кодировщика. Затем F по ветви восстановления напрямую передается в генератор изображения $O = \mathcal{D}(F)$, которое должно быть как можно ближе к входу I . Параллельно при переносе стиля i на I мы сворачиваем соответствующий банк фильтров \mathcal{K}_i с F , а затем вводим преобразованный признак $\tilde{F}_i (\tilde{F}_i = F \otimes \mathcal{K}_i)$ в \mathcal{D} для получения стилизованного результата $O_i = \mathcal{D}(\tilde{F}_i)$. Вышеуказанные две ветви обучаются разными способами, и соответственно разрабатываются разные функции потерь. Рассмотрим этот момент подробнее. Простая потеря MSE (mean square error, среднеквадратичная ошибка) между входным изображением I и O рассматривается как потеря отображения личности \mathcal{L}_J для ветви восстановления:

$$\mathcal{L}_J(I, O) = \|O - I\|^2. \quad (5.9)$$

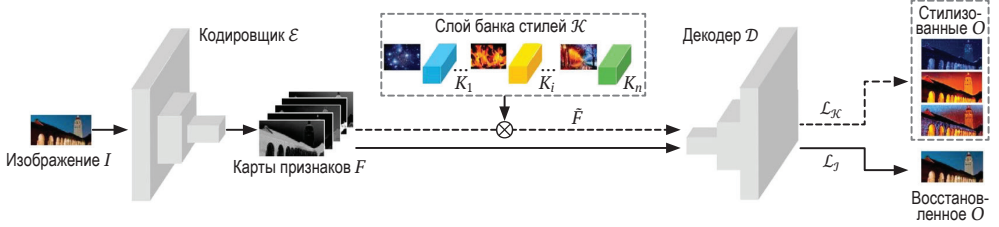


Рис. 5.10 ❖ Структура переноса разделенного стиля, предложенная в работе (Chen et al., 2017), которая состоит из одной ветви восстановления (внизу) и одной ветви стилизации (вверху). Контент предназначен для кодирования в ветке кодировщика и декодера, а стиль представлен набором банков фильтров в слое банка стилей. Источник: Chen et al. (2017)

В ветви стилизации, в соответствии с целевой функцией в работе (Johnson et al., 2016), в качестве потери стилизации \mathcal{L}_K используются потеря контента \mathcal{L}_c , потеря стиля \mathcal{L}_s и потеря регуляризации $\mathcal{L}_{tv}(O_i)$:

$$\mathcal{L}_K(I, \mathcal{S}_i, O_i) = \alpha \mathcal{L}_c(O_i, I) + \beta \mathcal{L}_s(O_i, \mathcal{S}_i) + \gamma \mathcal{L}_{tv}(O_i), \quad (5.10)$$

где I, \mathcal{S}_i, O_i – входной источник контента, источник стиля и результат стилизации (для i -го стиля) соответственно. $\mathcal{L}_{tv}(O_i)$ – регуляризатор полной вариации, используемый в работе (Johnson et al., 2016) для поощрения гладкости. И \mathcal{L}_c и \mathcal{L}_s используют одинаковое определение, приведенное в (Gatys et al., 2015):

$$\begin{aligned} \mathcal{L}_c(O, I) &= \sum_{l \in \{l_c\}} \|F^l(O) - F^l(I)\|^2; \\ \mathcal{L}_s(O, S) &= \sum_{l \in \{l_s\}} \|G(F^l(O)) - G(F^l(S))\|^2, \end{aligned} \quad (5.11)$$

где $G(X) = XX^T$.

Здесь F^l и G – карта объектов l -го слоя предварительно обученной сети VGG-16 и соответствующая матрица Грама, вычисленная из F . $\{l_c\}, \{l_s\}$ – это слои VGG-16, используемые для вычисления потери контента и потери сти-

ля. Поскольку ветвь восстановления предназначена для реконструкции исходного изображения контента, она гарантирует, что никакая информация о стиле не будет поглощена кодировщиком E и декодером D . В то же время, чтобы достичь желаемого стиля в ветви стилизации, вся информация о стиле принудительно передается в промежуточный слой банка стилей. Таким образом, контент и стиль явно отделены друг от друга.



Рис. 5.11 ❖ Процесс синтеза текстуры можно рассматривать как операцию свертки между изображением текстона и дельта-функцией в пространстве изображения, что побуждает нас выполнить передачу стиля в пространстве признаков путем свертки признака контента с соответствующим банком фильтров стиля

Результаты многостилевого переноса

Благодаря описанной выше раздельной схеме с двумя ветвями один стиль кодируется в одном конкретном наборе сверточных фильтров, и в одной сети можно одновременно изучать несколько стилей. Это более удобно для практического применения, чем предыдущие методы с одним стилем, которые обычно обучают одну независимую сеть для каждого стиля. Во время логического вывода для применения одного определенного стиля выбирается и применяется соответствующий набор фильтров. Как показано на рис. 5.12, разные стили очень хорошо разделены, и соответствующие результаты стилизации состоят только из их собственных паттернов стилей.



Рис. 5.12 ❖ Результаты применения нескольких разных стилей, которые одновременно изучены в одной сети. Видно, что разные стили хорошо разделены (каждый результат стилизации состоит только из собственных паттернов стилей) и изучены в соответствующих банках фильтров. Источник: Chen et al. (2017)

Восстановление элементов стиля

Чтобы лучше понять, как слой банка стилей представляет визуальный паттерн каждого стилизованного изображения, на примере рис. 5.13 показано восстановление элементов стиля из изученного банка фильтров. В частности, в результате стилизации выбираются два типа патчей: штрих (красная рамка) и текстура (зеленая рамка).

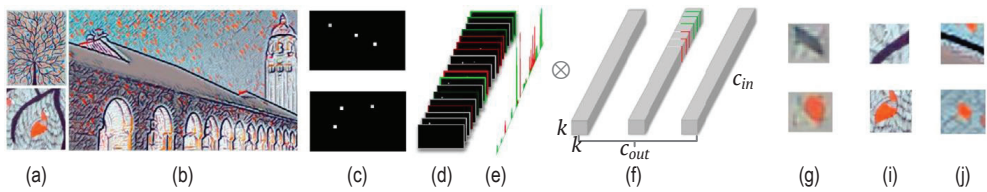


Рис. 5.13 ❖ Восстановление элементов стиля для двух патчей в примере стилизованного изображения. Источник: Chen et al. (2017)

Во-первых, все остальные области, кроме соответствующих положений выбранных участков, маскируются, как показано на (c, d), а распределение признаков этих участков визуализировано на (e). Можно заметить, что такие характерные отклики распределены редко, и отдельные пиковые отклики возникают лишь в некоторых каналах.

Затем в операции преобразования признаков учитываются только ненулевые каналы признаков и соответствующие каналы набора фильтров. Преобразованные признаки, наконец, передаются декодеру для получения восстановленных элементов стиля (g), которые визуальны похожи на патчи исходного стиля в (i) и патч стилизации в (j). Исходя из этого, мы можем предположить, что различные взвешенные комбинации каналов банка фильтров могут формировать разнообразные элементы стиля в одном изображении-источнике.

Чтобы понять, сколько различных элементов стиля изучается для каждого изображения стиля, мы берем большое шумовое изображение для аппроксимации распределения патчей контента и предоставляем сети возможность отображать различные патчи контента с разными элементами стиля. На рис. 5.14 видно, что стилизующая сеть изучила набор репрезентативных элементов для каждого изображения-источника стиля.



Рис. 5.14 ❖ Визуализация изученных сетью элементов стиля: для каждого случая левый элемент – это изображение-источник стиля, а два правых – результаты стилизации путем подачи двух разных шумовых изображений. Источник: Chen et al. (2020)

Важность разделения на две ветви

Чтобы показать, что наличие автокодировщика не приводит к утрате информации о стиле, а разделение на ветви имеет большое значение, был проведен еще один эксперимент путем удаления ветви восстановления на этапе обучения. При наличии одного входного изображения для вывода результата восстановления используются только кодер и декодер. Как показано на рис. 5.15, без ветви реконструкции на этапе обучения декодированное изображение (в центре) не может восстановить исходное входное изображение (слева), но, по-видимому, несет некоторую информацию о стиле. Для сравнения, когда используется ветвь восстановления, декодированное изображение идеально реконструирует входное изображение и очень похоже на него. Другими словами, в предложенной схеме с двумя ветвями вся информация о контенте сосредоточена только в части кодера и декодера, а информацию о стиле следует изучать на промежуточном уровне банка стилей.

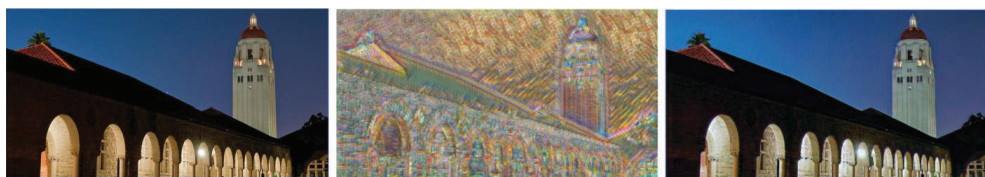


Рис. 5.15 ❖ Результат реконструкции изображения в механизме кодер–декодер с ветвью восстановления (справа) и без этой ветви (в центре) на этапе обучения. Очевидно, что наличие ветви восстановления помогает гарантировать, что информация о стиле не будет утрачена на этапе кодера и декодера, а реконструкция будет практически идеальной. Источник: Chen et al. (2017)

Преимущества разделения

Разделение контента и стиля может принести множество дополнительных преимуществ. Во-первых, это быстрое поэтапное обучение. В частности, чтобы включить новый стиль, нам не нужно переобучать всю сеть, что часто занимает много времени. Вместо этого, располагая обученной сетью с несколькими стилями, нам нужно только переобучить банк фильтров для вновь добавленного стиля, сохраняя при этом неизменные части кодировщика и декодера. Этот процесс сходится очень быстро, так как необходимо обучить только часть банка нового стиля. На практике это часто занимает всего несколько минут, что в десятки раз быстрее, чем переобучение всей сети. Как показано на рис. 5.16, частичное обучение может дать результаты стилизации, сравнимые с результатами полного переобучения всей сети с новыми стилями.

Второе преимущество заключается в возможности слияния стилей двумя разными способами: линейное слияние и слияние в зависимости от области. В случае линейного слияния, поскольку разные стили кодируются в разных банках фильтров $\{K_i^1, \dots, K_i^m\}$, мы можем линейно объединять несколько сти-

лей, объединяя банки фильтров в слой банка стилей. Затем объединенный банк фильтров используется для свертки с признаком контента F :

$$\tilde{F} = \left(\sum_{i=1}^m w_i * K_i \right) \otimes F \sum_{i=1}^m w_i = 1, \quad (5.12)$$

где m – количество стилей, K_i – банк фильтров стиля i . Затем \tilde{F} передается декодеру для получения окончательного результата стилизации. На рис. 5.17 показаны результаты такого линейного слияния двух разных стилей с разным весом слияния w_i .



Рис. 5.16 ❖ Разделение обеспечивает быстрое частичное изучение новых стилей. Для каждого случая слева и справа показаны результаты стилизации после дополнительного частичного обучения и нового обучения соответственно. Источник: Chen et al. (2017)



Рис. 5.17 ❖ Результаты стилизации линейной комбинацией двух наборов стилевых фильтров. Назначая разные веса слияния, мы соответствующим образом меняем пропорции каждого стиля. Источник: Chen et al. (2017)

Для слияния стилей в зависимости от конкретных областей предположим, что изображение разложено на n непересекающихся областей в пространстве признаков, а M_i обозначает маску каждой области, тогда карты признаков можно описать как $F = \sum_{i=1}^m (M_i \times F)$ и слияние стилей для конкретной области можно сформулировать следующим образом:

$$\mathcal{L}_{GR} = \begin{cases} \frac{1}{2} \|I_a - I'\|^2, & \text{если } I_s = I_a \\ \frac{\lambda}{2} \|I_a - I'\|^2 & \text{в ином случае} \end{cases}. \quad (5.13)$$

На рис. 5.18 показаны результаты слияния двух стилей для конкретных областей. Левый случай заимствует стили двух известных картин Пикассо и Ван Гога, а два правых стиля оба принадлежат Ван Гогу.

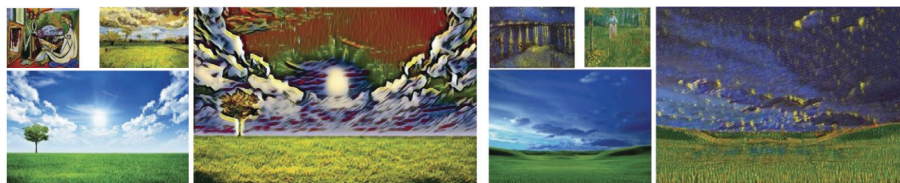


Рис. 5.18 ❖ Результаты слияния стилей в зависимости от области путем применения разных банков фильтров к разным областям изображения. Две левые картины принадлежат Пикассо и Ван Гоггу соответственно, а две правые принадлежат Ван Гоггу. Источник: Chen et al. (2020)

5.6.2. Разделение структуры и стиля

В отличие от описанной выше задачи переноса стиля, в которой семантическая структура контента и элементы стиля непосредственно определяются контентом и стилем входного изображения, в общем случае синтеза изображения более сложно добиться мелкоמודульного разделения. Например, в классическом методе StackGAN преобразования текста в изображение (Zhang et al., 2017) мы можем выбирать только стили в целом, вводя текстовые описания, но не можем добиться детального контроля как над структурой, так и над стилями. Здесь под структурой мы подразумеваем в первую очередь семантическую форму и позу, а стили обозначают мелкие визуальные паттерны/текстуры.

Для устранения упомянутого ограничения разработана новая каскадная генеративная модель FusedGAN, разделяющая структуры и стили методом частичного обучения с учителем (Bodla et al., 2018). Общая идея FusedGAN показана на рис. 5.19. Фактически здесь объединяются безусловная GAN для создания структуры приора и условная GAN для соответствия условию стиля из текстового описания. Ключевая идея проста: например, если вы хотите нарисовать птицу, наиболее очевидный подход – сначала набросать контур птицы с определенной позой и формой (создание структуры), а затем добавить детали мелкоמודульной текстурой (наложение стиля).

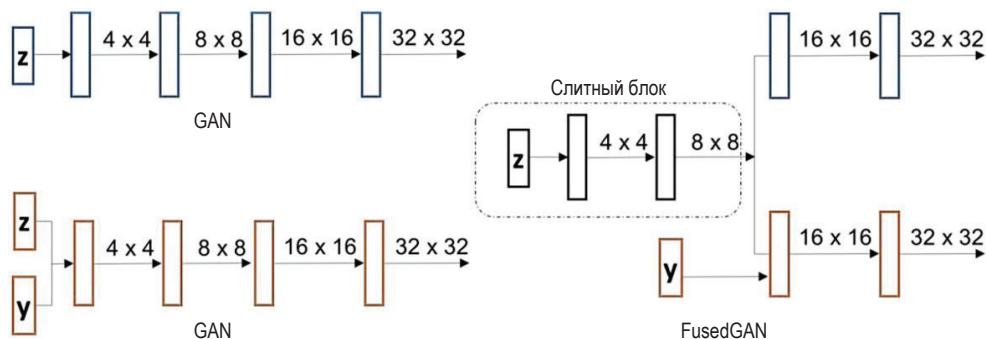


Рис. 5.19 ❖ Простая иллюстрация ключевой идеи FusedGAN, которая объединяет безусловную GAN и условную GAN. Источник: Bodla et al. (2018)

Разделение путем использования общей структуры приора

На основе вышеупомянутых принципов разработана изображенная на рис. 5.20 сквозная разделенная обучаемая структура, где синие и оранжевые блоки соответствуют безусловной и условной ветвям генерации изображений соответственно. В частности, безусловная ветвь состоит из сети генератора G_1 и сети дискриминатора D_u , которые обучаются таким же составительным способом, как и другие модели GAN. Чтобы обеспечить структуру для условной ветви генерации, сеть G_1 разбита на два модуля: G_s и G_u . Модуль G_s принимает вектор случайного шума z в качестве входных данных и после серии операций свертки и восстановления исходного разрешения (unsampling) генерирует приор M_s . Затем приор структуры M_s передается в G_u для создания конечного изображения после еще одной серии операций свертки и восстановления исходного разрешения.

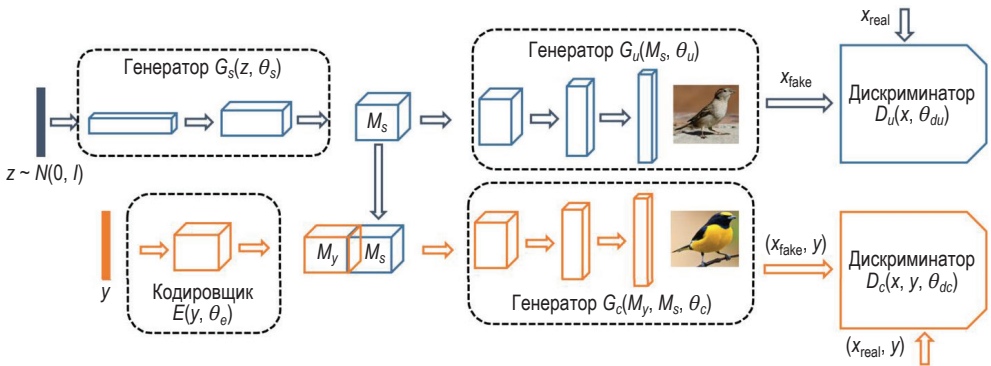


Рис. 5.20 ❖ Разделенная структура FusedGAN, где синие и оранжевые блоки представляют собой безусловный и условный конвейеры генерации изображений соответственно. Источник: Bodla et al. (2018)

В отличие от традиционной структуры условной генерации, которая часто принимает одно условие и один вектор случайного шума в качестве входных данных, сеть условного генератора G_c в FusedGAN вместо этого в качестве входных данных принимает приор M_s и вектор условия M_y . То есть ветвь безусловной генерации и ветвь условной генерации до M_s имеют одну и ту же структуру. Чтобы способствовать разделению структуры и стиля, ветви безусловной и условной генераций используют разные обучающие наборы данных, т. е. незамеченный набор данных для G_1 и набор данных, помеченный условными описаниями для G_c . Для разных задач вектор состояния M_y может иметь разный формат. Например, в задаче преобразования текста в изображение исходное текстовое описание сначала кодируется в представление y , а затем подается в кодировщик E для создания нового тензора условий M_y , как показано на рис. 5.20. Обратите внимание, что для получения разнообразных результатов выполняется условное дополнение выборки скрытого вектора \hat{c} из независимого гауссова распределения $N(\mu(y), \Sigma(y))$ вокруг представления текста, а затем \hat{c} пространственно повторяется до совпадения с пространственным измерением M_s для создания M_y .

Чтобы направлять обучение безусловной и условной ветвей генерации, дискриминатор D_u принимает в качестве входных данных сгенерированное изображение x_{uf} от G_u или реальное изображение x_r и пытается понять, является оно реальным или поддельным, в то время как дискриминатор D_c принимает в качестве входных данных сгенерированное изображение x_{cf} от G_c или реальное изображение x_r и соответствующее условие, чтобы гарантировать, что G_c генерирует изображения, соответствующие условию. На этапе обучения эти два конвейера обучаются порознь сквозным способом. Параметры модели обновляются путем оптимизации комбинированных целей GAN и CGAN, т. е.

$$\begin{aligned}\mathcal{L}_{G_u} &= \log D_u(G_u(z)), & \mathcal{L}_{D_u} &= \log D_u(x), & \mathcal{L}_{D_c} &= \log D_c(x, y); \\ \mathcal{L}_{G_c} &= \log D_c(G_c(M_y, M_s), y) + \lambda D_{KL}(N(\mu(y), \Sigma(y)) \parallel N(0, I),\end{aligned}\quad (5.14)$$

где z – выборочный вектор шума из нормального распределения $N(0, I)$. Подводя итог, можно сказать, что ключевой принцип разделения в этой работе заключается в использовании общего приора, но обучении двух ветвей с разными целями.

Во время логического вывода для генерации условного изображения сначала берется случайная выборка z , которая проходит через G_s для создания приора M_s . Затем M_s расходится на два пути – один ведет через генератор G_u для создания безусловного изображения x_{uf} . На втором пути входное текстовое описание подается в кодировщик E и извлекает выборку из гауссова распределения вокруг представления текста. Выходные данные E и M_s объединяются и прогоняются через G_c для создания условного изображения x_{cf} . Другими словами, на одном этапе вывода синтезируются два изображения: искомое условное изображение x_{cf} и безусловное изображение x_{uf} – фактически побочный продукт модели, который помогает анализировать и лучше понимать предложенную модель и результаты.

Результаты синтеза

Глядя на результаты синтеза на рис. 5.21, можно убедиться, что структура и стиль хорошо разделены. В левой части последняя строка отображает результаты безусловной генерации, а другие строки показывают результаты условной генерации с использованием того же структурного приора. Видно, что M_s способен успешно фиксировать и переводить значительный объем информации о строении птицы в результаты условного синтеза на основе различных текстовых описаний. Используя неизменный структурный приор, также легко удастся проводить интерполяцию между различными стилями. Для этого текстовые фрагменты t_1 и t_2 подаются в E для получения двух выборок из соответствующих им распределений Гаусса. Затем путем линейной интерполяции между ними выбираются восемь однородных отсчетов, так что первый отсчет соответствует t_1 , а последний – t_2 . Как показано в правой части рис. 5.21, эффект управляемой плавной интерполяции стилей вполне достигим.



Рис. 5.21 ❖ Синтезированные результаты FusedGAN. Левая часть представляет собой результаты синтеза различных стилей (строки) и структур (столбцы), а правая – результаты управляемой интерполяции различных стилей при фиксированной структуре. Источник: Bodla et al. (2018)

На рис. 5.22 дополнительно показаны результаты синтеза различных стилей с разным количеством мелких деталей. В частности, конкретное описание текстуры сначала загружается в E , а затем из гауссова распределения вокруг представления текста извлекаются пять выборок. Каждое текстовое описание может управлять результатами синтеза различных мелких деталей, используя один и тот же приор. Например, если взять второй ряд в левой части, хотя все птицы красные с черным крылом, у них разное количество черного на крыльях и длина хвоста.



Рис. 5.22 ❖ Синтезированные результаты различных стилей с помощью FusedGAN с разным количеством мелких деталей. Источник: Bodla et al. (2018)

5.6.3. Разделение личности и атрибутов

Третья значимая работа (Бао et al., 2018) посвящена типичной задаче генерации условного изображения «синтез лиц с сохранением личности». Мы имеем одно входное изображение I_{id} определенной личности и одно входное изображение I_a для извлечения атрибутов, а задача заключается в создании нового высококачественного изображения лица I' , которое относится к той же личности, что и I_{id} , но заимствует атрибуты I_a . Здесь атрибуты включа-

ют, помимо прочего, позу, эмоции, цвет кожи и фон. Это очень сложная задача, особенно когда соответствующая личность для лица не представлена в обучающем наборе, и основная проблема заключается в том, как разделить визуальные паттерны, связанные с личностью (например, форму носа и глаз), и визуальные паттерны, связанные с атрибутами (например, цвет кожи и форму рта при разных эмоциях).

Для решения этой проблемы было предложено множество схожих методов, таких как TP-GAN (Huang et al., 2017) и FF-GAN (Yin et al., 2017), которые могут синтезировать фронтальный вид данного изображения лица и DR-GAN (Tran et al., 2017), который может синтезировать различные положения лица. Однако они часто полагаются на полную аннотацию атрибутов, а список поддерживаемых типов атрибутов ограничен. Для сравнения, модель CVAEGAN (Bao et al., 2017) поддерживает различные изменения атрибутов, но не может синтезировать идентичное лицо вне набора обучающих данных.

Чтобы обеспечить большое разнообразие изменений атрибутов и открытых идентификаторов, была разработана новая схема разделенного синтеза лиц, изображенная на рис. 5.23 (Bao et al., 2018). Она содержит пять подсетей: сеть кодировщика личности E , сеть кодировщика атрибутов A , сеть генеративного синтеза G , вспомогательную сеть классификации C и сеть дискриминатора D . Функция сети кодировщика личности E и сети атрибутов A заключается в извлечении вектора личности $f_E(I_{id})$ из I_{id} и вектора атрибутов $f_A(I_a)$ из I_a соответственно. Путем рекомбинации $f_E(I_{id})$ и $f_A(I_a)$ сеть G генерирует новое изображение I' , которое соответствует личности I_{id} и атрибутам I_a . Вспомогательные сети C и D используются только во время обучения. В частности, C нужна для того, чтобы изображение I' отражало ту же личность, что и I_{id} , а сеть D побуждает G генерировать изображения более высокого качества, различая сгенерированное и реальное изображения в состязательном обучении.

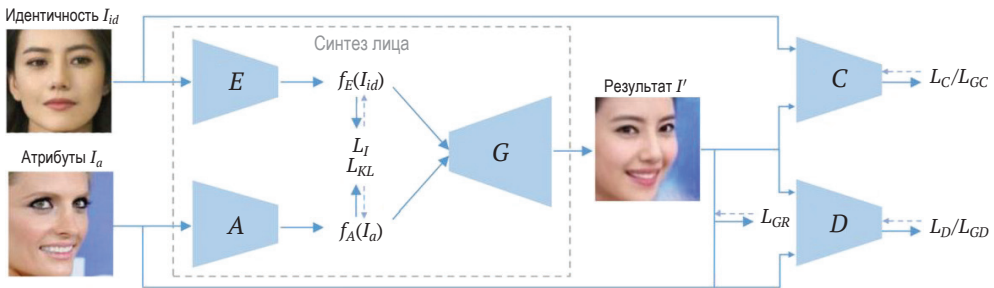


Рис. 5.23 ❖ Общая схема синтеза лица с сохранением личности, которая отделяет личность и атрибуты от входных изображений лица. Новое лицо I' генерируется путем рекомбинации информации о личности I_{id} и атрибутивной информации I_a . Источник: Bao et al. (2018)

Разделение

Несмотря на то что приведенный выше фреймворк основан на разделении, его обучение не является тривиальной задачей, потому что большинство

существующих наборов данных лиц имеют только аннотацию личности, но не аннотацию атрибутов. На самом деле иногда даже невозможно точно аннотировать некоторые атрибуты, например передний план и фон. Наше исследование посвящено разделению путем извлечения представления личности при помощи обучения с учителем и представления атрибутов при помощи обучения без учителя.

В частности, чтобы извлечь представление личности, кодировщик E формулируется как сеть распознавания лиц и обучается с помощью потери soft-тах на помеченном наборе данных лиц $\{I_i, y_i\}$, где y_i – метка личности изображения I_i . Во время обучения, чтобы различать разных людей, E обучается представлять сходные признаки для изображений с одинаковыми идентификаторами и разные признаки для изображений с разными идентификаторами. Отклик последнего пулингового слоя E принимается в качестве вектора личности I_{id} .

Для получения представления атрибута разработана простая и эффективная стратегия обучения, использующая потерю реконструкции и потерю расхождения KL. В частности, во время обучения I_{id} и I_a выбираются случайным образом, поэтому могут относиться к одному и тому же или к разным изображениям. В обоих случаях для реконструкции изображения атрибута I_a требуется результирующее изображение I' , но с разными весами потерь.

Формально потеря реконструкции равна

$$\mathcal{L}_{GR} = \begin{cases} \frac{1}{2} \|I_a - I'\|^2, & \text{если } I_s = I_a \\ \frac{\lambda}{2} \|I_a - I'\|^2 & \text{в ином случае} \end{cases}, \quad (5.15)$$

где λ – вес потерь при реконструкции для случая $I_s \neq I_a$. В частности, когда изображение личности I_{id} совпадает с изображением атрибута I_a , синтезированное изображение I должно быть таким же, как I_{id} или I_a . Поскольку для каждой личности в обучающем наборе есть много изображений лиц I_a , их вектор личности для этих изображений почти одинаков, и единственное возможное различие будет заключаться в векторе атрибутов. Следовательно, требование, чтобы реконструированные изображения были такими же, как эти разные изображения лиц, вынуждает сеть A кодировщика атрибутов правильно выучить различные представления атрибутов. В случае когда изображение личности I_{id} и изображение атрибута I_a различны, хотя трудно точно предсказать, как должен выглядеть реконструированный результат, мы можем ожидать, что реконструкция будет приблизительно аналогична изображению атрибута I_a , например в отношении фона, общего освещения и позы. Следовательно, для сохранения атрибутов потере реконструкции исходного пикселя назначается относительно небольшой вес ($\lambda = 0,1$).

Помимо вышеупомянутой потери реконструкции, дополнительно используется потеря расхождения KL для регуляризации вектора атрибутов с соответствующим priorом $P(z) \sim N(0, 1)$. Это ограничивает вектор атрибута, чтобы он не содержал чрезмерного количества информации о личности, и помогает сети кодировщика атрибутов изучить лучшее представление. Для

заданного входного изображения лица, которое служит источником атрибута, сеть A выведет среднее значение μ и ковариацию скрытого вектора. Тогда потеря расхождения KL определяется следующим образом:

$$\mathcal{L}_{KL} = \frac{1}{2}(\mu^T \mu + \text{sum}(\exp(\epsilon) - \epsilon - 1)). \quad (5.16)$$

Во время обучения используется прием репараметризации для выборки вектора атрибутов с использованием $z = \mu + r \odot \exp(\epsilon)$, где $r \sim N(0, I)$ – случайный вектор, а \odot представляет поэлементное умножение.

Асимметричная настройка

После извлечения вектора личности $f_I(I_{id})$ и вектора атрибутов $f_A(I_a)$ выполняется их конкатенация в скрытом пространстве $z' = [f_I(I_{id}), f_A(I_a)]$ и результат передается в сеть G для синтеза нового изображения лица. Подобно обычным GAN, генерирующая сеть G играет в минимаксную игру для двух игроков с дискриминаторной сетью D , т. е. D пытается отличить реальные настроечные данные от синтезированных данных, в то время как G пытается обмануть сеть D . Конкретно, сеть D пытается минимизировать функцию потерь

$$\mathcal{L}_D = -\mathbb{E}_{I_a \sim p_r}[\log D(I_a)] - \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \quad (5.17)$$

Однако если сеть G напрямую пытается максимизировать D как традиционная GAN, процесс обучения будет нестабильным. Это связано с тем, что на практике распределения «настоящих» и «поддельных» изображений могут не пересекаться друг с другом, особенно на раннем этапе настройки. Следовательно, дискриминаторная сеть D может идеально их разделить, что вызовет исчезновение градиента. Чтобы решить эту проблему, для обучения генератора используется потеря сопоставления парных признаков, как в CVAE-GAN. В частности, если предположить, что $f_D(\cdot)$ – это признаки промежуточных слоев D , потери сопоставления парных признаков определяются следующим образом:

$$\mathcal{L}_{GD} = \frac{1}{2} \|f_D(I') - f_D(I_a)\|_2^2. \quad (5.18)$$

То есть эта потеря максимально сближает признаки реального и сгенерированного изображений. По умолчанию выходной признак последнего полносвязного слоя D используется как f_D .

Точно так же для достижения цели сохранения личности (I' относится к той же личности, что и I_{id}) используется аналогичная потеря сопоставления парных признаков, чтобы заставить I' и I_{id} иметь аналогичные представления признаков в сети классификации лиц C :

$$\mathcal{L}_{GC} = \frac{1}{2} \|f_C(I') - f_C(I_{id})\|_2^2. \quad (5.19)$$

Здесь вход последнего полносвязного уровня сети C используется как признак f_C . На практике сеть C и сеть E имеют общие параметры и инициализи-

руются предварительно обученной сетью классификации лиц для ускорения сходимости.

Стратегия обучения без учителя

Синтез лиц для личностей, которые отсутствуют в обучающем наборе, является сложной задачей, поскольку требуется, чтобы генеративная сеть охватывала как внутриличностные (intraperson), так и межличностные (interperson) вариации. Существующие общедоступные наборы данных с помеченными личностями часто имеют ограниченный размер и не содержат экстремальных поз или освещения, поэтому мы собрали из Flickr и Google один миллион изображений лиц с большими вариациями и разнообразием. Затем мы провели обучение без учителя, чтобы помочь обученному генератору обобщить ранее не встречавшиеся личности. В частности, собранные неразмеченные изображения можно использовать либо в качестве изображения личности I_{id} , либо в качестве изображения атрибутов I_a . При использовании изображения в качестве источника атрибута I_a весь процесс обучения остается неизменным. При использовании в качестве источника личности I_{id} , поскольку изображения не имеют метки класса, они не участвуют в обучении E и C . Эмпирически мы обнаружили, что эти неразмеченные данные могут увеличить внутриклассовые и межклассовые различия в распределении лиц, тем самым улучшая разнообразие синтезированных лиц, например внося более значительные изменения в позы и выражения.

Результаты синтеза

Для демонстрации эффективности описанного выше фреймворка отдельного синтеза лиц на рис. 5.24 и 5.25 изображены результаты синтеза, в котором используются изображения, чьи личности как встречаются, так и не встречаются в обучающем наборе соответственно. Результаты свидетельствуют о том, что обученная сеть может отделить компоненты личности и атрибутов и очень хорошо изучить соответствующие визуальные паттерны как для закрытых, так и для открытых наборов данных.

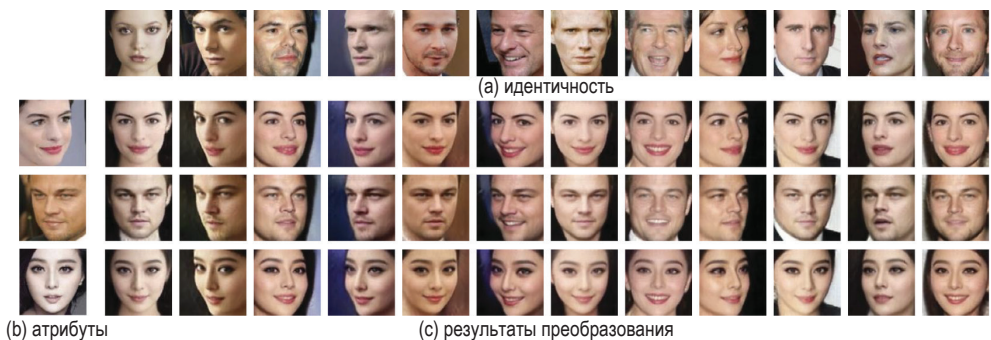


Рис. 5.24 ❖ Результаты синтеза лиц с сохранением личности с использованием изображений, чьи личности присутствуют в обучающем наборе данных. Видно, что предложенный метод может хорошо разделять визуальные паттерны, связанные с идентификацией и атрибутами, а затем перекомпоновывать их в окончательные результаты. Источник: Bao et al. (2018)

Разделение также обеспечивает непрерывное изменение атрибутов в сгенерированных изображениях путем настройки скрытого вектора, так называемого *морфинга атрибутов*. В частности, для пары изображений I_{a1} и I_{a2} сеть атрибутов A сначала используется для извлечения их векторов атрибутов z_{a1} и z_{a2} соответственно, а затем с помощью линейной интерполяции может быть получен ряд векторов атрибутов z , т. е. $z = \alpha z_{a1} + (1 - \alpha) z_{a2}$, $\alpha \in [0, 1]$. На рис. 5.26 представлены результаты морфинга атрибутов лица, где путем выбора подходящей пары изображений-источников атрибутов постепенно изменяются положение, эмоции или освещение.



Рис. 5.25 ❖ Результаты синтеза лиц с сохранением личности с использованием изображений, личности которых не появляются в обучающем наборе данных, что демонстрирует сильную способность к обобщению. Источник: Bao et al. (2018)



Рис. 5.26 ❖ Результаты морфинга лица с использованием невидимых идентичностей между двумя атрибутами с точки зрения позы, эмоций и изменения освещения соответственно. Источник: Bao et al. (2018)

Применение синтеза

Приложения проверки лиц на основе глубокой сети широко используются в системах наблюдения и контроля доступа. Располагая изображениями двух лиц, предварительно обученная модель классификации лиц сначала извлекает их признаки. Затем, если расстояние между признаками меньше порогового значения, два лица рассматриваются как относящиеся к одной и той же личности. Тем не менее недавние исследования показывают, что

глубокие нейронные сети уязвимы для злонамеренных имитаций, которые обманывают сеть, добавляя определенные малозаметные возмущения к исходным изображениям. В частности, предположив, что два лица I_1 и I_2 относятся к разным личностям, мы можем подобрать незаметные возмущения r таким образом, что вышеупомянутой системой проверки лиц $I_1 + r$ будет рассматриваться как лицо, совпадающее с I_2 .

Этот процесс часто формулируется как задача оптимизации:

$$\min \|r\|_2^2$$

$$\text{так, что } \|f_C(I_1 + r) - f_C(I_2)\|_2^2 < \tau, \quad (5.20)$$

где f_C – признак, извлеченный из предварительно обученной сети, а τ – заданный порог.

На рис. 5.27 (а) и (с) – два входа I_1 и I_2 , а (b) – сгенерированное враждебное изображение $I_1 + r$. Несмотря на то что враждебные изображения имеют сходные признаки с другими лицами, если мы реконструируем изображение из признака с использованием предложенного фреймворка, это даст нам изображение совершенно другого человека (е). Очевидно, что враждебное изображение и его реконструкция явно различаются. Основываясь на этом наблюдении, описанный выше фреймворк можно использовать для обнаружения враждебных изображений путем сравнения личностей на исходных изображениях и результатов реконструкции.



Рис. 5.27 ❖ Обнаружение враждебных образцов в системах проверки лиц: (а) исходное изображение, (b) является враждебным образцом, который направлен на то, чтобы ввести сеть проверки в заблуждение относительно личности, показанной в (с). (d), (e) и (f) – результаты реконструкции, выполненной нашим фреймворком. Они показывают, что хотя враждебный образец выглядит так же, как и исходное изображение, результаты реконструкции выглядят по-разному. Источник: Bao et al. (2018)

Взяв в качестве примера набор данных LFW, для каждой из 3000 пар различных личностей мы сгенерировали два враждебных изображения путем генерации ложных изображений друг друга, что дает в сумме 6000 враждебных изображений. Здесь используются четыре разных порога τ : [0,4, 0,6, 0,8, 1]. При этом у нас есть 6000 исходных изображений и их реконструкций. Затем признак LBP входного изображения и его реконструированное изображение извлекаются и объединяются вместе. Наконец, линейный SVM обучается как бинарный классификатор. Результаты показаны в табл. 5.1. Если расстояние между признаками меньше 0,4, можно достичь точности обнаружения 92,41 %.

Таблица 5.1. Точность обнаружения враждебных изображений при различных порогах расстояния между признаками

Порог	1,0	0,8	0,6	0,4
Точность, %	76,73	82,58	87,18	92,41

5.7. ЗАКЛЮЧЕНИЕ

Изучение и моделирование визуальных паттернов являются фундаментальной основой визуального интеллекта. С помощью безусловной или условной структуры генерации изображений глубокие генеративные модели пытаются восстановить низкоразмерную структуру целевых визуальных моделей в пространстве представлений. В этой главе мы обсудили, как использовать глубокие генеративные модели для достижения более управляемого синтеза визуальных паттернов посредством создания условного изображения. Мы утверждаем, что ключом к достижению такого управляемого синтеза паттернов является разделение визуального представления, когда различные управляющие факторы тем или иным способом разделяют в скрытом пространстве представлений. Затем на примере трех исследований мы продемонстрировали, как добиться разделения для синтеза паттернов при обучении без учителя или со слабым обучением путем введения индуктивной предпосылки с точки зрения структуры сети и стратегии обучения.

В классических генеративных моделях зачастую различные смешанные факторы явно моделируются как взаимодействующие случайные процессы. Этот вид явного моделирования не был хорошо изучен в глубоких генеративных моделях. Возможно, было бы полезно изучить, как мы можем ввести такие явные и структурированные представления в обучение глубоких генеративных сетей, что может привести к более объяснимым глубоким моделям.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Arjovsky M., Chintala S., Bottou L., 2017. Wasserstein gan. arXiv preprint. arXiv: 1701.07875.
- Bao J., Chen D., Wen F., Li H., Hua G., 2017. Cvae-gan: fine-grained image generation through asymmetric training. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2745–2754.
- Bao J., Chen D., Wen F., Li H., Hua G., 2018. Towards open-set identity preserving face synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6713–6722.
- Besag J., 1974. Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B, Methodological 36, 192–225.
- Blake A., Zisserman A., 1987. Visual Reconstruction. MIT Press.
- Bodla N., Hua G., Chellappa R., 2018. Semi-supervised fusedgan for conditional image generation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 669–683.

- Chen D., Liao J., Yuan L., Yu N., Hua G.*, 2017a. Coherent online video style transfer. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1105–1114.
- Chen D., Yuan L., Liao J., Yu N., Hua G.*, 2017b. Stylebank: an explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1897–1906.
- Chen D., Yuan L., Liao J., Yu N., Hua G.*, 2018. Stereoscopic neural style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6654–6663.
- Chen D., Fan Q., Liao J., Aviles-Rivero A., Yuan L., Yu N., Hua G.*, 2020a. Controllable image processing via adaptive filterbank pyramid. *IEEE Transactions on Image Processing* 29, 8043–8054.
- Chen D., Yuan L., Hua G.*, 2020b. Deep style transfer. *Computer Vision: A Reference Guide*, 1–8.
- Chen D., Yuan L., Liao J., Yu N., Hua G.*, 2020c. Explicit filterbank learning for neural image style transfer and image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Chen X., Duan Y., Houthoofd R., Schulman J., Sutskever I., Abbeel P.*, 2016. Info-gan: interpretable representation learning by information maximizing generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2172–2180.
- Cooper D. B.*, 1979. Maximum likelihood estimation of Markov-process blob boundaries in noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 372–384.
- Cross G. R., Jain A. K.*, 1983. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25–39.
- Dempster A. P., Laird N. M., Rubin D. B.*, 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B, Methodological* 39, 1–38.
- Do M. N., Vetterli M.*, 2003. The finite ridgelet transform for image representation. *IEEE Transactions on Image Processing* 12, 16–28.
- Doersch C.*, 2016. Tutorial on variational autoencoders. *arXiv preprint. arXiv:1606.05908*.
- Fan Q., Chen D., Yuan L., Hua G., Yu N., Chen B.*, 2018. Decouple learning for parameterized image operators. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 442–458.
- Fan Q., Chen D., Yuan L., Hua G., Yu N., Chen B.*, 2019. A general decoupled learning framework for parameterized image operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Fu K. S.*, 1982. *Syntactic Pattern Recognition*. Prentice-Hall.
- Gatys L. A., Ecker A. S., Bethge M.*, 2015. A neural algorithm of artistic style. *arXiv preprint. arXiv:1508.06576*.
- Geman S., Geman D.*, 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 721–741.
- Ghahramani Z., Beal M. J.*, 2001. Graphical models and variational methods. In: *Graphical Models and Variational Methods*. In: *Neural Information Processing Series*. MIT Press, Cambridge, MA.

- Girshick R.*, 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448.
- Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.*, 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680.
- Grenander U.*, 1976, 1976–1981. Lectures in pattern theory i, ii and iii.
- Guo C. E., Zhu S. C., Wu Y. N.*, 2003. Modeling visual patterns by integrating descriptive and generative methods. *International Journal of Computer Vision* 53, 5–29.
- He K., Zhang X., Ren S., Sun J.*, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.
- He M., Chen D., Liao J., Sander P. V., Yuan L.*, 2018. Deep exemplar-based colorization. *ACM Transactions on Graphics (TOG)* 37, 1–16.
- Hinton G. E.*, 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation* 14, 1771–1800.
- Hoyer P. O., Hyvärinen A.*, 2002. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research* 42, 1593–1605.
- Hua G.*, 2020. Deep generative models. In: *Computer Vision: a Reference Guide*. Springer.
- Huang R., Zhang S., Li T., He R.*, 2017. Beyond face rotation: global and local perception gan for photorealistic and identity preserving frontal view synthesis. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2439–2448.
- Hyvärinen A.*, 1999. Survey on independent component analysis.
- Hyvärinen A., Oja E.*, 2000. Independent component analysis: algorithms and applications. *Neural Networks* 13, 411–430.
- Isola P., Zhu J. Y., Zhou T., Efros A. A.*, 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134.
- Johnson J., Alahi A., Fei-Fei L.*, 2016. Perceptual losses for real-time style transfer and super-resolution. In: *European Conference on Computer Vision*. Springer, pp. 694–711.
- Kambhatla N., Leen T. K.*, 1997. Dimension reduction by local principal component analysis. *Neural Computation* 9, 1493–1516.
- Kingma D. P., Welling M.*, 2014. Auto-encoding variational Bayes. In: Bengio, Y., LeCun, Y. (Eds.), *Interactional Conference on Learning Representation*. <http://dblp.uni-trier.de/db/conf/iclr/iclr2014.html#KingmaW13>.
- Kong H., Wang L., Teoh E. K., Li X., Wang J. G., Venkateswarlu R.*, 2005. Generalized 2d principal component analysis for face image representation and recognition. *Neural Networks* 18, 585–594.
- Lee A. B., Mumford D., Huang J.*, 2001. Occlusion models for natural images: a statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision* 41, 35–59.
- Liu D., Hua G., Chen T.*, 2010. A hierarchical visual model for video object summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 2178–2190.

- Locatello F., Bauer S., Lucic M., Rätsch G., Gelly S., Schölkopf B., Bachem O.*, 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In: Proc. of the 36th International Conference on Machine Learning. Long Beach, CA.
- Lu X., Katz A., Kanterakis E., Li Y., Zhang Y., Caviris N.*, 1992. Image analysis via optical wavelet transform. *Optics Communications* 92, 337–345.
- Ma L., Sun Q., Georgoulis S., Van Gool L., Schiele B., Fritz M.*, 2018. Disentangled person image generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 99–108.
- Manat S., Zhang Z.*, 1993. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing* 12, 3397–3451.
- Metz L., Poole B., Pfau D., Sohl-Dickstein J.*, 2016. Unrolled generative adversarial networks. arXiv preprint. arXiv: 1611.02163.
- Mirza M., Osindero S.*, 2014. Conditional generative adversarial nets. arXiv preprint. arXiv:1411.1784.
- Mumford D. B., Shah J.*, 1989. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*.
- Ng A., et al.*, 2011. Sparse Autoencoder. CS294A. Lecture Notes, vol. 72, pp. 1–19.
- Poggio T., Torre V., Koch C.*, 1985. Computational vision and regularization theory. *Nature* 317, 314–319.
- Ren S., He K., Girshick R., Sun J.*, 2015. Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99.
- Roweis S., Ghahramani Z.*, 1999. A unifying review of linear Gaussian models. *Neural Computation* 11, 305–345.
- Shackleton M.*, 1994. Learned deformable templates for object recognition. In: IEE Colloquium on Genetic Algorithms in Image Processing and Vision, IET, p. 7.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.*, 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
- Tan Z., Chai M., Chen D., Liao J., Chu Q., Yuan L., Tulyakov S., Yu N.*, 2020a. Michigan: multi-input-conditioned hair image generation for portrait editing. *ACM Transactions on Graphics (TOG)* 39, 95.
- Tan Z., Chen D., Chu Q., Chai M., Liao J., He M., Yuan L., Hua G., Yu N.*, 2020b. Semantic image synthesis via efficient class-adaptive normalization. arXiv preprint. arXiv:2012.04644.
- Terzopoulos D.*, 1983. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing* 24, 52–96.
- Tran L., Yin X., Liu X.*, 2017. Disentangled representation learning gan for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1415–1424.
- Ulyanov D., Lebedev V., Vedaldi A., Lempitsky V. S.*, 2016. Texture networks: feed-forward synthesis of textures and stylized images. In: ICML, p. 4.
- Van de Wouwer G., Scheunders P., Van Dyck D.*, 1999. Statistical texture characterization from discrete wavelet representations. *IEEE Transactions on Image Processing* 8, 592–598.

- Van den Oord A., Kalchbrenner N., Espeholt L., Vinyals O., Graves A., et al., 2016. Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems, pp. 4790–4798.
- Vincent P., Larochelle H., Bengio Y., Manzagol P. A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103.
- Wan Z., Zhang B., Chen D., Zhang P., Chen D., Liao J., Wen F., 2020a. Bringing old photos back to life. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2747–2757.
- Wan Z., Zhang B., Chen D., Zhang P., Chen D., Liao J., Wen F., 2020b. Old photo restoration via deep latent space translation. arXiv preprint. arXiv:2009.07047.
- Wang Y., Yao H., Zhao S., 2016. Auto-encoder based dimensionality reduction. *Neurocomputing* 184, 232–242.
- Wang Z., She Q., Ward T. E., 2019. Generative adversarial networks in computer vision: a survey and taxonomy. arXiv preprint. arXiv:1906.01529.
- Weng J., Weng C., Yuan J., Liu Z., 2018. Discriminative spatio-temporal pattern discovery for 3d action recognition. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1077–1089.
- Xie X., Sudhakar R., Zhuang H., 1994. On improving eye feature extraction using deformable templates. *Pattern Recognition* 27, 791–799.
- Yan X., Yang J., Sohn K., Lee H., 2016. Attribute2image: conditional image generation from visual attributes. In: European Conference on Computer Vision. Springer, pp. 776–791.
- Yin X., Yu X., Sohn K., Liu X., Chandraker M., 2017. Towards large-pose face frontalization in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3990–3999.
- Yuan J., 2011. Discovering Visual Patterns in Image and Video Data: Concepts, Algorithms, Experiments Paperback. VDM Verlag Dr. Müller.
- Yuille A. L., 1991. Deformable templates for face recognition. *Journal of Cognitive Neuroscience* 3, 59–70.
- Zhai J., Zhang S., Chen J., He Q., 2018. Autoencoder and its various variants. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 415–419.
- Zhang H., Xu T., Li H., Zhang S., Wang X., Huang X., Metaxas D. N., 2017. Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915.
- Zhao G., Yuan J., Hua G., 2013. Topical video object discovery from key frames by modeling word co-occurrence prior. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'2013). Portland, OR.
- Zhu J. Y., Park T., Isola P., Efros A. A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232.
- Zhu S. C., 2003. Statistical modeling and conceptualization of visual patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 691–712.
- Zhu S. C., Wu Y., Mumford D., 1998. Filters, random fields and maximum entropy (frame): towards a unified theory for texture modeling. *International Journal of Computer Vision* 27, 107–126.

Глава 6

Глубокое распознавание лиц с использованием полных и частичных изображений

Автор главы:

Хассан Угайл, Центр визуальных вычислений,
Университет Брэдфорда, Брэдфорд, Великобритания

Краткое содержание главы:

- методы и практические приемы определения, обучения и тестирования модели на основе глубокого обучения;
- примеры использования моделей на основе глубокого обучения для распознавания лиц по полному и частичному изображениям;
- современное положение дел в области распознавания лиц, а также проблемы с глубоким обучением, помогающим распознавать лица.

6.1. ВВЕДЕНИЕ

Распознавание лиц – одно из самых захватывающих и известных приложений глубокого обучения, применяемых в области визуальных вычислений. Оно не только демонстрирует мощь глубокого обучения, но и выявляет некоторые проблемы, связанные с использованием нейросетевых моделей в виде черного ящика. Решение этих проблем окажет прямое влияние на нашу повседневную жизнь.

Компьютерное распознавание лиц по-прежнему сопряжено со многими проблемами, которые не свойственны человеческому восприятию. Человеку часто бывает достаточно увидеть кого-то лишь мельком, чтобы навсегда запомнить его лицо (Young, Burton, 2018). Так получается потому, что мозг моментально выделяет и запоминает важные детали, касающиеся человека.

Затем, когда знакомое лицо предстает в различных контекстах, мозг легко и быстро сопоставляет изображения «до» и «после» без потребности в какой-либо значимой новой информации. А когда речь идет о компьютерном зрении, малейшие изменения внешнего вида лица могут существенно ухудшить способность идентифицировать человека.

В компьютерном зрении существует множество алгоритмов машинного обучения, специально разработанных для применения в распознавании лиц. Эти алгоритмы представляют собой методы, основанные на обучении без учителя или, наоборот, использующие обучение с учителем и основанные на идее, что кто-то может выбрать часть данных с известными метками (предоставленными оператором) или передать алгоритму в качестве обучающего набора. Например, *анализ главных компонент* (principal component analysis, PCA) был изобретен еще в 1905¹ г., а сегодня фактически превратился в модель машинного обучения без учителя, широко используемую для уменьшения размерности сложных данных и сжатия изображений (Jolliffe, 2002). Также широко распространен метод кодирования лиц в наборе данных в «пространстве лица». В 1991 году для построения алгоритмов распознавания лиц был предложен базовый алгоритм Eigenface (Turk, Pentland, 1991). Этот алгоритм извлекает наиболее важные детали данных в виде *собственных векторов*², соответствующих наибольшим *собственным значениям*, представляющим изменения в пространстве лица. В области распознавания лиц наиболее интересна концепция *усредненного лица* (average face), и в некотором смысле она лежит в основе распознавания человеческих лиц. Попытки использовать усредненное лицо в качестве инструмента компьютерного распознавания лиц часто встречаются в научных публикациях, например (Elmahmudi, Ugail, 2019). Также широко используется метод *локальных бинарных шаблонов* (local binary pattern, LBP), который представляет собой простой, но мощный подход к классификации текстур. Согласно этому методу изображение делят на разные области, чтобы извлечь признаки из каждой области отдельно. Эти признаки впоследствии используются для облегчения классификации в задачах распознавания лиц (Kas et al., 2020).

В обычных вычислениях данный алгоритм представляет собой группу явно запрограммированных команд, используемых машиной для решения задачи. Подходы машинного обучения позволяют обучать алгоритмы с использованием входных данных и использовать статистический анализ для вывода значений, попадающих в определенную область. Иными словами, машинное обучение позволяет компьютерам создавать модели из примеров, автоматизируя процесс принятия решений на основе ввода данных.

Глубокое обучение (LeCun et al., 2015) – это механизм, с помощью которого машинный алгоритм может обучаться на образцах данных. Он направлен на получение оптимальной конфигурации модели, позволяющей получать из набора входных данных желаемые результаты. Обученные модели широко

¹ По другим данным, метод главных компонент был изобретен Карлом Пирсоном в 1901 г. – *Прим. перев.*

² Понятия собственного вектора и собственного значения были введены в разделе 1.6.4. – *Прим. перев.*

применяются для решения сложных задач обработки и анализа изображений. В результате глубокое обучение, вероятно, стало стандартом де-факто в современных системах распознавания лиц.

До эпохи глубокого обучения большая часть алгоритмов распознавания лиц была основана на методах обработки изображений, содержащих только два или три уровня вычислений, таких как фильтрация, гистограммы и кодирование признаков. У этих методов был общий существенный недостаток – они решали только один аспект проблемы распознавания лиц за счет других. Например, хотя *метод фильтрации Габора* (Dora et al., 2017) может улучшить распознавание лиц при различных условиях освещения, этот же метод плохо работает с разными выражениями и ракурсами лица. Поэтому исследователи пытались отыскать последовательный и целостный метод для решения большинства проблем, мешающих распознаванию лиц.

Революционный прорыв произошел в 2012 году, когда было убедительно показано, что глубокое обучение способно решить многие проблемы, с которыми в то время сталкивалось компьютерное распознавание лиц. В то время модель глубокого обучения AlexNet выиграла конкурс ImageNet, продемонстрировав, что она стабильно и с большим отрывом опережает конкурентов в распознавании изображений. С тех пор глубокое обучение движется по восходящей траектории и приближается к уровню человеческой способности распознавания лиц. Например, в 2014 году набор данных DeepFace на маркированных лицах в дикой природе (LFW) показал, что он может достичь точности на уровне человека, точные цифры составляют 97,35 % для DeepFace и 97,533 % для людей (Taigman et al., 2014).

6.1.1. Модели глубокого обучения

Глубокое обучение направлено на эмуляцию и изучение сложных структур, скрытых в наборах данных с использованием нескольких обучающих слоев функциональных единиц (нейронов). Этот подход имитирует строение и работу нервной системы человека. Модели глубокого обучения изучают закономерности в очень сложных данных с помощью последовательностей из нескольких искусственных нейронов, манипулируя параметрами межнейронных связей для достижения желаемого результата. Поэтому модель глубокого обучения состоит из нейронной сети с несколькими скрытыми слоями. Она предназначена для имитации принятия решений человеком при решении сложных задач распознавания. На практике модели глубокого обучения обычно принимают форму сверточных нейронных сетей (CNN) (LeCun et al., 2015).

Структура CNN

По сути, CNN представляет собой набор отдельных *персептронов* (искусственных нейронов), образующих сеть нейронов, соединенных между собой для обеспечения параллельной обработки сигналов в распределенной сетевой структуре. Для управления взаимодействием между персептронами используются настраиваемые веса. Ключевым компонентом CNN являются

скрытые слои, которые размещаются между входом и выходом сети. Скрытые слои позволяют присваивать нелинейные веса входным данным и отправлять результаты на выход, т. е. применять нелинейные математические функции к определенным частям сети для получения желаемого конечного результата. Например, в случае задачи распознавания лиц один скрытый слой можно настроить для идентификации цветов входного изображения, другой – для идентификации физических признаков и т. д. Сеть, состоящая из этих слоев, может затем распознать и классифицировать лицо на входном изображении.

Типичная CNN состоит из нескольких слоев, которые можно разделить на три широкие категории. Это слои *свертки* (convolution, CONV), *субдискретизации*, или *пулинга* (pooling, POOL), и *полносвязные* слои (fully connected, FC). Обычно комбинация этих слоев устроена определенным образом с единственной целью преобразования входных данных сети в полезное *представление*, которое дает выходной прогноз, как показано на рис. 6.1.

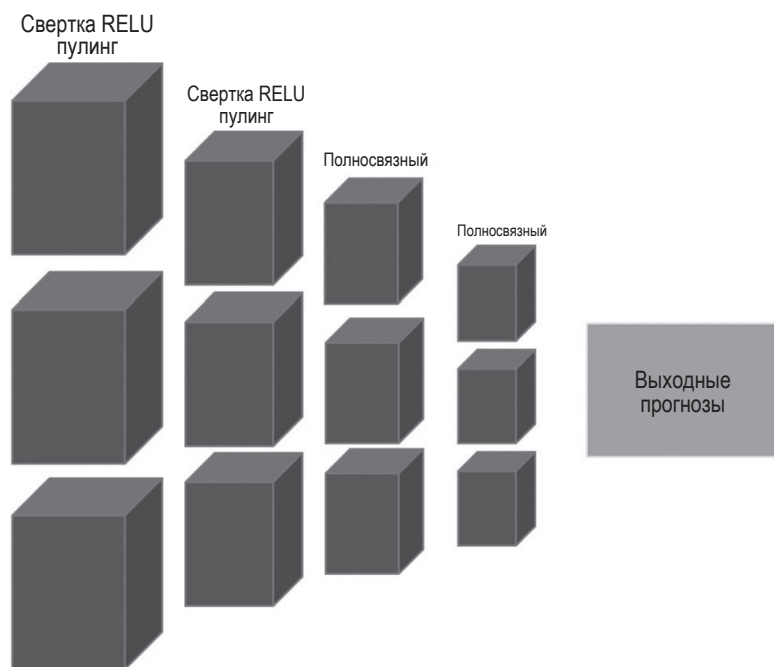


Рис. 6.1 ❖ CNN состоит из нескольких слоев, включая слои свертки, пулинга и полностью связанные слои

Слой свертки получил свое название от математического оператора свертки. Этот слой вычисляет скалярное произведение между весами нейронов и небольшой областью входного пространства. Нейроны организованы в виде набора двумерных *фильтров*, или *ядер*, расширяющих размерность входных данных. Следовательно, они имеют трехмерную структуру. Во время прямого прохода каждое ядро сворачивается по ширине и высоте входного

пространства для создания двумерной *карты признаков* (feature map), как показано на рис. 6.2. Эти карты признаков являются выходными данными операции свертки при каждой пространственной операции. По сравнению с нейронной сетью с прямым распространением, эти фильтры представляют собой нейроны, которые активируются, когда они сталкиваются с визуальными признаками, такими как края объекта. Как обсуждалось ранее, CNN используют локальную связь для уменьшения сложности. Следовательно, каждый нейрон связан с локальной областью, пространственный размер которой определяется размером фильтра, известным как *рецептивное поле нейрона*, а ее глубина равна глубине входных данных.

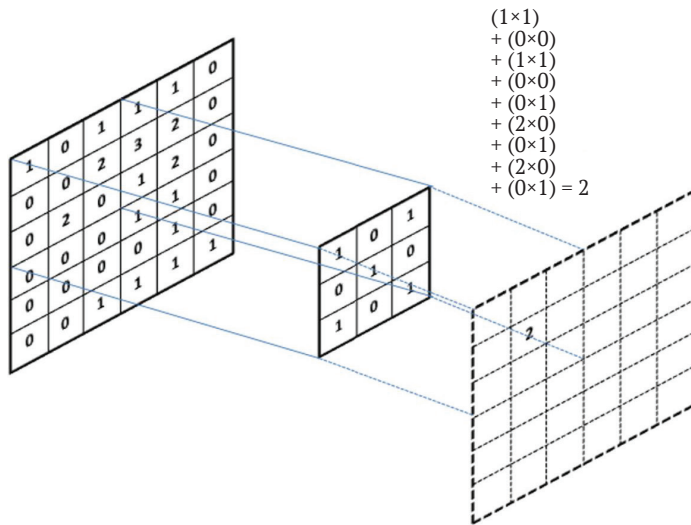


Рис. 6.2 ❖ Иллюстрация операции свертки. Во время прямого прохода каждое ядро сворачивается по ширине и высоте входного пространства для создания 2D-карты признаков

Следовательно, в случае входного изображения $256 \times 256 \times 3$ и рецептивного поля 3×3 каждый нейрон в слое CONV будет иметь в общей сложности $3 \times 3 \times 3 = 27$ связей и 1 параметр смещения. Очевидно, что связность является пространственно локальной, но полной по входной глубине. Впоследствии размер карты признаков (то есть выходных данных) вычисляется с использованием трех гиперпараметров – *глубины* (depth), *паддинга* (padding) и *страйда* (stride). Глубина соответствует числу развернутых фильтров. Чем больше фильтров, тем больше объем извлекаемой информации, поскольку каждый фильтр учится искать определенный признак. Страйд S определяет шаблон, используемый для перемещения фильтра по входным данным, т. е. $S = 1$ означает, что фильтр должен перемещаться на один пиксель за раз. Паддинг определяет количество нулевых пикселей, размещенных вокруг входного пространства, чтобы сохранить постоянный пространственный размер вывода. Можно вычислить пространственный размер вывода, используя уравнение

$$\frac{I - F + 2P}{S} + 1 = 0, \quad (6.1)$$

где I – пространственный размер входных данных, F – размер фильтра, P – заполнение нулями (падинг), а S – страйд (Dumoulin and Visin, 2018). Следовательно, если взять входное изображение размера $224 \times 224 \times 3$ и предположить, что нейроны имеют рецептивное поле размером 3×3 , глубину $K = 64$, одиночный страйд $S = 1$ и падинг $P = 1$, получаем $(224 - 3 + 2)/1 + 1 = 224$. Это означает, что вывод этого конкретного слоя CONV будет иметь размер $224 \times 224 \times 64$. Следовательно, слой должен состоять из $224 \times 224 \times 64 = 3\,211\,264$ нейронов, каждый из которых имеет $3 \times 3 \times 3 = 27$ весов и 1 смещение. Интересно, что вместо $3\,211\,264 \times 27$ весов и $3\,211\,264$ смещений благодаря концепции *общих весов* (weight sharing) все нейроны на одном срезе могут иметь одинаковый вес и смещение. Следовательно, количество весов и смещений резко сокращается до 1728 и 64 соответственно.

Как упоминалось ранее, в нейронных сетях функция активации играет важную роль в привнесении нелинейности в выходной сигнал нейрона. Введение этой нелинейности делает нейронную сеть *универсальным аппроксиматором функций*, тем самым давая ей возможность справляться с различными типами отношений. Наиболее эффективной и часто используемой функцией активации для CNN является спрямленный линейный блок (ReLU). Он представляет собой поэлементное применение нулевой пороговой функции $f(x) = \max(0, x)$, где x – вход нейрона. По сравнению с другими функциями активации, CNN с ReLU обучаются в несколько раз быстрее. Это связано с простотой вычисления функции ReLU и ее градиента. Обратите внимание, что слой активации не получает на входе дополнительные параметры. Кроме того, он не меняет размерность ввода. В архитектуре сети слои активации размещаются после каждого слоя CONV. Кроме того, они также развертываются в сетях с более чем одним полносвязным слоем, за исключением последнего такого слоя.

Слои пулинга обычно вставляются между последовательными слоями свертки. Их основная функция состоит в том, чтобы последовательно сокращать количество параметров и, следовательно, уменьшать вычислительную сложность сети за счет уменьшения пространственного размера карт признаков. Слои пулинга суммируют выходные данные соседних нейронов. Для каждого 2D-среза карты признаков наиболее часто применяется так называемый *тах-пулинг* (maximum pooling), который обычно берет максимум каждой области 2×2 , таким образом отбрасывая 75 % активаций, как показано на рис. 6.3 (Gholamalinezhad, Khosravi, 2020).

Таким образом, операция пулинга не вводит новых параметров. Наоборот, она приводит к сокращению первого и второго измерений карты признаков. Операция принимает два параметра: страйд S и пространственное измерение F . Следовательно, операция пулинга уменьшает размер карты признаков с размеров $W_1 \times H_1 \times D$ до $W_2 \times H_2 \times D$. Здесь W_2 и H_2 вычисляются по следующим формулам:

$$W_2 = \frac{W_1 - F}{S + 1}, \quad H_2 = \frac{H_1 - F}{S + 1}. \quad (6.2)$$

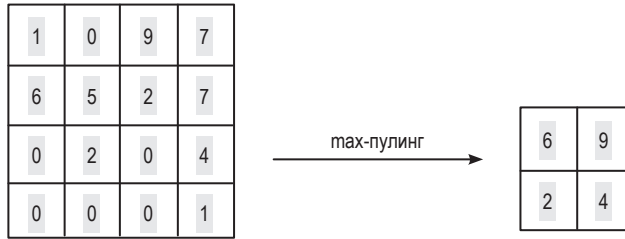


Рис. 6.3 ❖ Слой пулинга в действии. Пулинг помогает уменьшить количество параметров и, следовательно, снизить вычислительную сложность сети

Интересно, что эта операция вводит трансляционную инвариантность по отношению к упругим искажениям. Полносвязный слой FC имеет нейроны, которые полностью связаны с активацией предыдущего слоя, и, в отличие от слоев CONV и POOL, слой FC является двумерным. Обычно полносвязные слои выполняют вывод предсказанных сетью меток/классов. Следовательно, FC обычно является последним уровнем сети. В работе, победившей в конкурсе ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 г. (Krizhevsky et al., 2012), использовались три слоя FC, и с тех пор это является типичным решением. Достаточно очевидно, что сведение трехмерных карт признаков в конце вычислений дает нам возможность интерпретировать изученные пространственно-инвариантные признаки.

Наиболее обычная схема, используемая исследователями, начинается со слоя ввода изображения и заканчивается слоем FC (решение), между ними находятся повторяющиеся стеки слоев CONV-ReLU, за которыми следуют слои POOL, а затем несколько слоев FC-ReLU. Эту многослойную структуру можно описать математически:

$$\text{ВХОД} \Rightarrow N\text{M}(\text{CONV} \Rightarrow \text{ReLU}) \Rightarrow \text{POOL} \Rightarrow K(\text{FC} \Rightarrow \text{ReLU}) \Rightarrow \text{FC}, \quad (6.3)$$

где N , M и K – положительные вещественные параметры. Обычно количество слоев CONV-ReLU, которые размещены перед POOL, находится в диапазоне $0 < N < 4$, а число комбинаций переменных M и K больше 1.

Методы обучения CNN

Существует три основных способа развертывания CNN: обучение сети с нуля, точная настройка существующей модели или использование готовых признаков CNN. Последние два подхода называются *переносом обучения*, или *трансферным обучением* (transfer learning). Поскольку обучение CNN с нуля с использованием алгоритма обратного распространения предполагает автоматическое изучение миллионов параметров, этот подход требует огромного количества данных и, как следствие, нуждается в большой вычислительной мощности. Кроме того, процедура включает в себя настройку нескольких гиперпараметров. Поэтому на практике сеть редко обучают с нуля.

Точная настройка включает перенос весов первых n слоев, полученных из исходной (опорной) сети, в целевую сеть, а затем завершение обучения с использованием нового набора данных. Таким образом, целевая сеть обучается

с использованием нового набора данных для конкретной задачи, обычно отличной от задачи исходной сети. Точная настройка обычно используется, когда новый набор данных умеренно велик (от десятков до сотен тысяч обучающих размеченных образцов) и сильно отличается от набора данных, используемого для обучения исходной сети. Использование весов исходной сети для инициализации помогает алгоритму обратного распространения, что приводит к относительно быстрому автоматическому обучению более конкретным признакам.

В ситуациях, когда набор данных довольно мал, скажем несколько сотен, даже точная настройка весов может привести к переобучению. Однако, поскольку CNN эффективно изучают общие признаки изображения, можно напрямую использовать обученную сеть в качестве средства извлечения фиксированных признаков. Следовательно, признаки из новых данных извлекаются путем проецирования их на активации определенного слоя предварительно обученной сети. После этого изученные представления передаются в простые классификаторы для решения поставленной задачи. Этот подход, известный как *извлечение готовых признаков* (off-the-shelf feature extraction), принес многообещающие результаты (Weiss et al., 2016; Day and Khoshgoftaar, 2017).

Самый простой способ борьбы с проблемой переобучения, с которой сталкиваются глубокие нейронные сети, – это увеличение объема обучающих данных (*дополнение данных*, data augmentation). Обычно дополнение данных реализуют путем искусственного увеличения размера набора данных с помощью различных методов, таких как изменение ориентации изображения путем отражения (что создает зеркальное изображение) и поворот исходных изображений, что впоследствии сеть рассматривает как новые изображения. Этот прием гарантирует, что алгоритм обучения выведет признаки из данных с разной ориентацией.

Наборы данных для экспериментов с глубоким распознаванием лиц

Существует ряд наборов данных о лицах, которые можно использовать для обучения и тестирования моделей глубокого распознавания лиц. Здесь мы подробно расскажем о некоторых из них.

Набор данных LFW (Labeled Faces in the Wild) (Huang et al., 2008) представляет собой большой набор изображений лиц, предназначенный для тестирования способности распознавания лиц в смоделированных сценариях. Все изображения были собраны из интернета и состоят из спектра вариаций выражения лица, позы, возраста, освещения и разрешения. База данных LFW содержит изображения 5749 объектов, всего около 13 000 изображений. Сами изображения в наборе данных имеют разнообразные и значительные фоновые помехи.

База данных лиц YouTube (He et al., 2018) состоит из видеороликов лиц с различным освещением, положением и возрастом. База данных специально разработана для изучения и анализа алгоритмов распознавания лиц в видеороликах. Она содержит более 3000 видеороликов, на которых сняты 1500 человек. Видео были загружены с YouTube.

Набор данных FEI (Thomaz, Giralaldi, 2010) содержит 200 изображений бразильских студентов и преподавателей с равным количеством мужчин

и женщин. Для каждого субъекта есть 14 изображений, общее количество изображений в наборе данных составляет 2800. Разрешение изображений – 640×480 пикселей, и все изображения сделаны в цвете на однородном белом фоне. Субъектам от 19 до 40 лет, и набор данных содержит изображения, отображающие различные выражения и положения лица.

6.2. КОМПОНЕНТЫ СИСТЕМЫ ГЛУБОКОГО РАСПОЗНАВАНИЯ ЛИЦ

По сути, современная система распознавания лиц, основанная на глубоком обучении, состоит из трех частей. Это *обнаружение* лица, с помощью которого система находит лицо на изображении, *обработка*, с помощью которой лицо обрезается и часто нормализуется, и *распознавание*, в ходе которого алгоритм глубокого обучения используется для классификации или сопоставления лица. Этот процесс изображен на рис. 6.4.

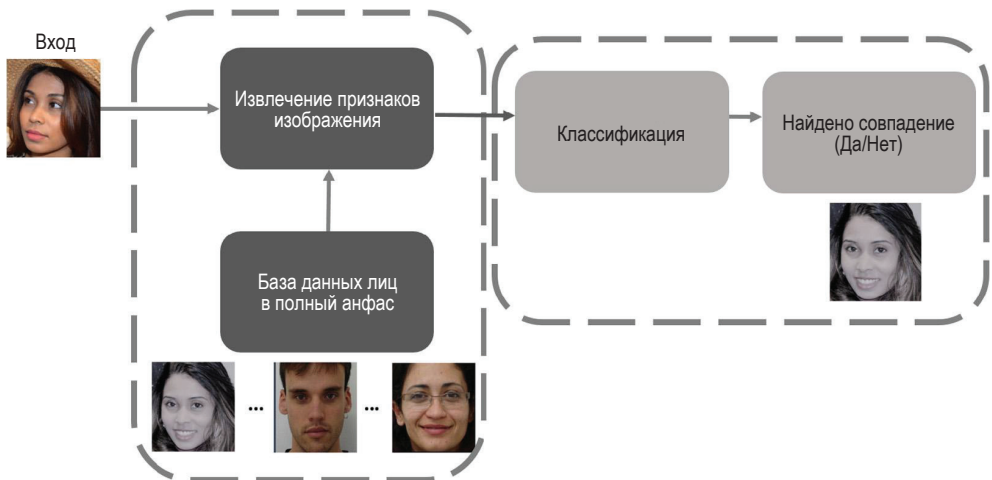


Рис. 6.4 ❖ Компоненты системы глубокого распознавания лиц

Хотя глубокое обучение эффективно применяется для решения большинства проблем распознавания объектов, для задач распознавания лиц нам по-прежнему приходится прибегать к этапу извлечения лица из сцены на этапе обработки, чтобы гарантировать, что влияние ракурса, освещения, мимики и окклюзии сведено к минимуму.

Следовательно, систему глубокого распознавания лиц в обобщенном виде можно описать следующим образом:

$$M[F(I_i), F(I_j)], \quad (6.4)$$

где I_i и I_j – два сравниваемых изображения лица, F – признаки из модели CNN, а M определяет критерии совпадения. После обычно очень длительного про-

цесса обучения с массивными наборами данных и с наблюдением за соответствующими функциями потерь определяются оптимальные слои, из которых необходимо извлечь и сравнить признаки. Сам процесс сравнения может быть осуществлен с использованием мер расстояния, таких как евклидово расстояние, косинусное сходство (CS) и машины опорных векторов (SVM).

Сегодня распознавание лиц на практике в основном выполняется с помощью моделей глубокого обучения, и, как упоминалось ранее, нам есть из чего выбирать. Далее мы обсудим некоторые примеры, чтобы дополнительно объяснить процесс распознавания лиц на основе глубокого обучения и проблемы, которые требуют особого внимания.

Важной частью любой системы распознавания лиц, основанной на глубоком обучении, являются признаки, полученные из обученной модели CNN (Liu et al., 2017). Пользователь может выбирать из нескольких архитектур моделей. К ним относятся AlexNet (Krizhevsky et al., 2012), GoogleNet (Szegedy et al., 2015), ResNet (He et al., 2015) и VGGNet (Parkhi et al., 2015).

6.2.1. Пример обученной модели CNN для распознавания лиц

Как упоминалось в предыдущих главах, существует несколько способов развертывания CNN. К ним относятся обучение сети с нуля, тонкая настройка существующей модели или использование готовых признаков CNN из предварительно обученной модели. Последнее называется переносом обучения.

Важно подчеркнуть, что для обучения CNN с нуля требуется огромное количество данных, что часто является сложной задачей. Например, для обучения модели FaceNet потребовались миллионы лиц и сотни часов вычислительного времени (Schroff et al., 2015). В свою очередь, точная настройка включает перенос весов первых нескольких слоев, полученных из базовой сети, в целевую сеть. Затем целевая сеть может быть обучена с использованием нового набора данных.

Хорошим примером предварительно обученной модели CNN в контексте распознавания лиц является модель VGG-F (Chatfield et al., 2016), разработанная исследовательской группой Oxford Visual Geometry Group. Эта модель была обучена на большом наборе данных из 2,6 млн изображений лиц более чем 2,6 тыс. человек. Архитектура VGG-F состоит из 38 слоев, начиная с входного уровня и заканчивая выходным. На вход должно поступать цветное изображение с разрешением 224×224 , и в качестве шага предварительной обработки из входного изображения обычно вычисляется среднее значение.

В общей сложности VGG-F содержит тринадцать сверточных слоев, каждый из которых имеет специальный набор гибридных параметров. Каждая группа сверточных слоев содержит 5 слоев max-пулинга и 15 слоев ReLU. После них идут три слоя FC, а именно FC6, FC7 и FC8. Первые два имеют 4096 каналов, тогда как FC8 имеет 2622 канала, которые используются для классификации 2622 идентичностей. Последний слой – это классификатор softmax, представляющий вероятность того, что изображение принадлежит к определенному классу. Архитектура VGG-F представлена на рис. 6.5.

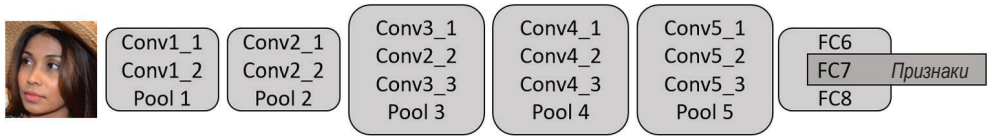


Рис. 6.5 ❖ Архитектура модели VGG-F. Модель содержит 13 сверточных слоев, каждый слой имеет специальный набор гибридных параметров

Далее мы покажем, как модель VGG-F, предварительно обученную для извлечения признаков, можно использовать для кодирования черт лица и как можно использовать меру косинусного сходства или линейные меры SVM для классификации для эффективного распознавания лиц по неполным изображениям.

Извлечение признаков

Заданное входное изображение X_0 можно представить как тензор $X_0 \in R^{HWD}$, где H – высота изображения, W – ширина, а D – цветовые каналы. Предварительно обученный слой L CNN может быть выражен как ряд функций $g_L = f_1 \rightarrow f_2 \rightarrow \dots \rightarrow f_L$.

Пусть X_1, X_2, \dots, X_n будут выходами каждого слоя в сети. Затем выходные данные i -го промежуточного слоя могут быть вычислены по функции f_i и изученным весам w_i таким образом, что $X_i = f_i(X_{(i-1)}; w_i)$.

Как мы знаем, CNN изучают признаки на этапе обучения и позже используют их для классификации изображений. Каждый сверточный слой изучает разные признаки. Например, один слой может изучить такие признаки, как края и цвета изображения, в то время как более сложные признаки могут быть изучены на более глубоких уровнях. Например, выход сверточного слоя включает в себя множество двумерных массивов, которые называются каналами. В VGG-F есть 37 слоев, 13 из которых – свертки, а остальные слои представляют собой смесь функций ReLU, пулинга, softmax и полносвязных слоев. Однако после применения слоя conv5_3 с 512 фильтрами размера 3×3 можно извлечь признаки для классификации. Изучая активации этого слоя, можно получить основные признаки, как изображено на рис. 6.6, где представлена выборка признаков.

Чтобы решить, какой слой в модели VGG-F лучше всего использовать для извлечения признаков лица, необходимо провести ряд экспериментов методом проб и ошибок. Тесты показывают (Elmahmudi, Ugail, 2019), что, как правило, наиболее эффективны слои с 34 по 37. Часто наилучшие результаты дает слой 34. Следует отметить, что этот слой является полносвязным и располагается в конце CNN, поэтому извлеченные признаки представляют все лицо.

Признаки из слоя 34 – это результаты, полученные из полносвязного слоя FC7 после применения ReLU6, которые представляют собой вектор с размерностью 4096. Предположение о том, что слой 34 является наиболее подходящим, основано на результатах проведения ряда тестов распознавания лиц с использованием полного фронтального изображения лица как для обучения, так и для тестирования. Результаты свидетельствуют, что точность распознавания может достигать 100 %.

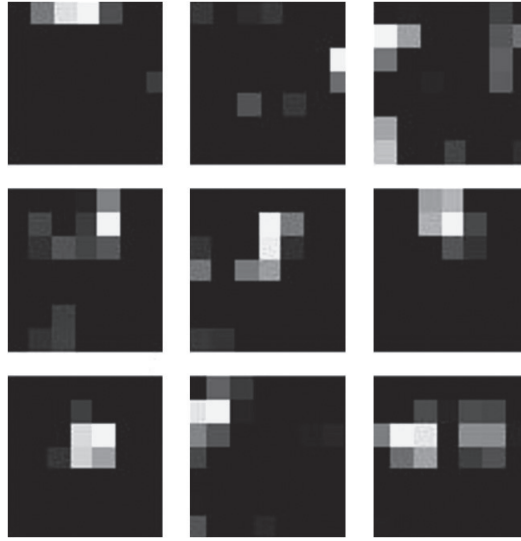


Рис. 6.6 ❖ Визуализация признаков в VGG-F, полученная из слоя conv5_3 после ввода изображения лица

Классификация признаков

Одной из задач классификации является построение краткой модели распределения меток классов по прогнозируемым признакам. Существует несколько методов такой классификации, среди них наиболее известны деревья решений, k -ближайшие соседи (k -nearest neighbors, kNN) и SVM.

SVM – это алгоритм машинного обучения с учителем, который можно использовать как для задач бинарной классификации, так и для задач распределения объектов по нескольким классам (мультиклассификация). SVM работает путем построения гиперплоскостей, проходящих через многомерное пространство признаков для разделения данных на классы, и поиска «зазора» между гиперплоскостями. Чем больше зазор между гиперплоскостями, тем ниже ожидаемая ошибка классификации. Очевидно, что алгоритм SVM ориентирован прежде всего на решение задач бинарной классификации. Для решения задачи мультиклассификации обычно используют линейный SVM в сочетании с методом «один против одного» (one-vs-one, OVO), также известным как *парная классификация*. Декомпозиция OVO дает $\frac{n(n-1)}{2}$ бинарных классификаторов для n классов. Затем, используя метод *кодов исправления ошибок* (error correction codes, ECC), принимают решение, как можно комбинировать различные классификаторы.

Если у нас есть обучающий набор данных (x_i, y_i) , мы можем использовать линейный SVM таким образом, что:

$$\min \frac{1}{2} \|w\|^2 + C \sum_i^N \max(0, 1 - y_i w^T x_i), \quad w \in R^d, \quad (6.5)$$

где w – вектор весов, N – количество классов, а C – параметр компромисса между ошибкой и «зазором».

Кроме того, для классификации можно также использовать *косинусное подобие*¹ (cosine similarity, CS). Это мера сходства между двумя ненулевыми векторами, которая использует пространство скалярных произведений для измерения косинуса угла между двумя векторами. Для вычисления косинусного подобия можно использовать формулу евклидова скалярного произведения:

$$a \cdot b = |a||b| \cos \theta, \quad (6.6)$$

где a и b – два вектора, а θ – угол между ними. Используя длину $|x|$, которая совпадает с евклидовой нормой или евклидовой длиной вектора $x = [x_1, x_2, x_3, \dots, x_n]$, мы можем вычислить косинусное подобие S следующим образом:

$$|x| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}; \quad (6.7)$$

$$S = \cos \theta = \frac{A \cdot B}{|A||B|} = \frac{\sum_i^N A_i B_i}{\sqrt{\sum_i^N A_i^2} \sqrt{\sum_i^N B_i^2}}, \quad (6.8)$$

где A и B – два вектора.

Для классификации субъекта можно вычислить CS и найти минимальное «расстояние» между тестовым изображением лица $test_{im}$ и обучающим изображением $training_{im}^n$ с использованием уравнения (6.9), такое, что:

$$M_{CS} = \min(CS(test_{im}, training_{im}^n)), \quad (6.9)$$

где im – номер изображения, n – общее количество изображений в обучающей выборке.

6.3. Распознавание лиц с использованием полных изображений лица

Что касается глубокого обучения, хорошо обученная модель CNN часто обеспечивает превосходное распознавание лиц по полному изображению. В этом разделе мы покажем, как можно использовать типичную предварительно обученную модель для задачи сопоставления и проверки лиц.

Модель FaceNet (Schroff et al., 2015) создана на основе архитектуры модели GoogLeNet и предварительно обучена для эффективного распознавания лиц. Для задачи распознавания лиц она использует изображения списка людей в наборе данных вместе с данными нового человека или людей, которых

¹ Также известен как коэффициент Оцуки–Оттаи, геометрический или косинусный коэффициент. – Прим. перев.

нужно распознать. Ключевым элементом архитектуры FaceNet является создание представления определенной размерности из изображения лица с заданным разрешением. Входное изображение пропускают через глубокую архитектуру CNN, которая имеет полностью связанный слой в конце. На выходе получается 128 признаков, которые не обязательно могут быть визуально понятными человеку. Затем для распознавания сеть может рассчитать расстояние между отдельными признаками каждого из представлений. Для вычисления расстояния между представлениями можно использовать такие метрики, как квадрат ошибки или абсолютная ошибка.

Обычные модели FaceNet используют два типа архитектуры. Это архитектура Цейлера и Фергуса (Zeiler and Fergus, 2014) и модель Inception в стиле GoogLeNet (Szegedy et al., 2015). Основной идеей в обучении FaceNet является концепция *триплетных потерь* (triplet loss) для выявления сходств и различий между классами лиц в 128-мерном представлении. Если дано представление $E(x)$ изображения в пространстве признаков R^n , FaceNet смотрит на квадрат расстояния L_2 между изображениями лиц, причем это значение мало для изображений с одной и той же идентичностью и велико для разных идентичностей.

На рис. 6.7 изображена общая архитектура модели FaceNet. Важным элементом модели является функция триплетных потерь. Часто в обычных моделях глубокого обучения функция потерь пытается сопоставить все изображения лица одной и той же идентичности с одной точкой в R^n . Функция триплетных потерь пытается отличить каждую *пару* изображений лица одного человека от всех остальных, тем самым обеспечивая строгое различие лиц. Таким образом, функция триплетной потери, выбранная в этой модели, гарантирует, что изображение конкретного человека будет *ближе* ко всем другим изображениям этого человека, чем любое другое изображение в наборе данных. Эта идея проиллюстрирована на рис. 6.8. Процесс обучения предполагает, что мы выбираем из набора данных случайное изображение – *якорь*. Нам нужно обучить модель так, чтобы расстояние между этим изображением-якорем и другим изображением того же человека (положительные экземпляры) было меньше, чем расстояния между якорем и изображениями, не принадлежащими этому человеку (отрицательные экземпляры).



Рис. 6.7 ❖ Общая архитектура модели FaceNet. Ключевым аспектом модели является возможность классифицировать изображение лица, используя 128-мерное пространство признаков

Модель FaceNet была обучена на наборе из 100 млн образцов лиц с более чем 8 млн личностей. Путем экспериментов было обнаружено, что оптимальное представление для лиц имеет размерность 128, т. е. каждое лицо различается с использованием 128 извлеченных из изображения признаков. Также были проведены эксперименты по варьированию обучающих данных

и было установлено, что после определенного момента увеличение числа обучающих выборок почти не увеличивает точность, т. е. первые несколько десятков миллионов выборок еще могут повысить точность, но увеличение набора до сотни миллионов почти не повысит точность распознавания.

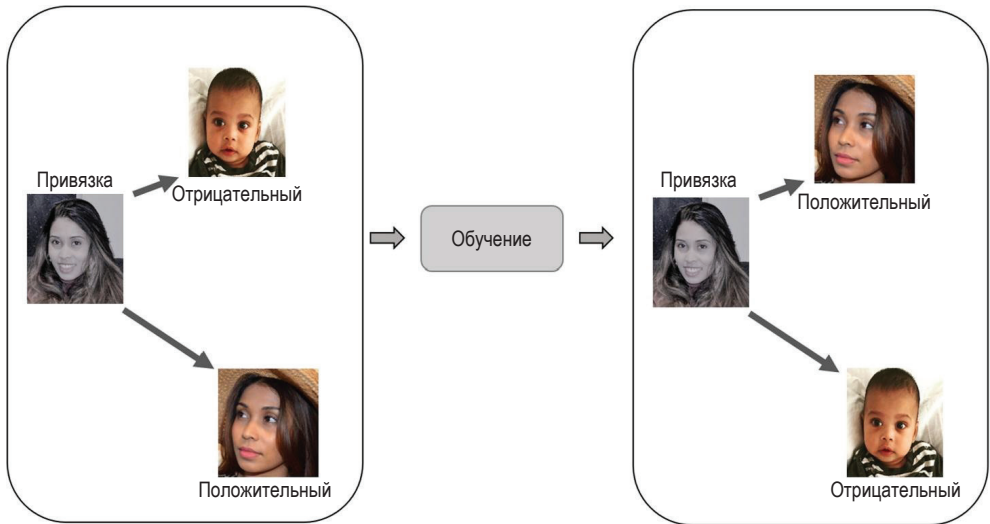


Рис. 6.8 ❖ Иллюстрация того, как FaceNet использует обучение на основе триплетной потери. Функция триплетной потери оценивает отличие каждой пары лиц одного человека от всех остальных, тем самым обучая модель точно различать лица

6.3.1. Проверка подоби́я с использованием модели FaceNet

Одним из основных преимуществ модели FaceNet является то, что она позволяет достичь очень высокой точности классификации, используя более простое представление, состоящее всего из 128 признаков. Эксперименты, проведенные с лицами как из набора LFW, так из базы данных YouTube Faces, показывают, что точность распознавания очень высока: в наборе данных LFW достигается точность распознавания 98,87 %, а в базе данных YouTube Faces – 95,18 %. Также важно подчеркнуть, что оба набора данных содержат лица, снятые при разном освещении, положении, окклюзии и в разном возрасте. Несмотря на это, точность распознавания FaceNet впечатляет.

На рис. 6.9 показано, как можно использовать модель FaceNet для определения степени сходства между лицами. В этом примере 128 признаков каждого лица извлекаются с использованием модели FaceNet. Затем можно воспользоваться мерой косинусного подоби́я для вычисления расстояния между признаками каждого лица и изображением лица в центре. Результаты показывают, что точность сопоставления лиц в этом случае оказывается пре-

восходной. Стоит отметить также, что в экспериментах с моделью FaceNet подобие двух лиц на уровне 75 % или выше обычно считается полным совпадением.



Рис. 6.9 ❖ Пример проверки подобия. Для сравнения портретов королевы используется мера косинусного подобия на основе 128 признаков модели FaceNet. Центральное изображение используется как образец для сравнения

6.4. ГЛУБОКОЕ РАСПОЗНАВАНИЕ НЕПОЛНЫХ ИЗОБРАЖЕНИЙ ЛИЦА

Если проанализировать результаты исследований, проведенных в области распознавания лиц с использованием глубокого обучения, становится ясно, что многие современные алгоритмы обеспечивают точность распознавания лиц на уровне человека, когда на изображении лицо расположено строго в анфас (фронтально). Например, рассмотренная выше модель FaceNet обеспечивает впечатляющий уровень точности распознавания с использованием фронтальных изображений лица. Однако на практике полное изображение лица может быть недоступно ни в качестве эталона, ни в качестве экземпляра для сравнения. В этом разделе мы обсудим, как методы, основанные на глубоком обучении, могут развиваться дальше, поскольку модели можно научить успешно распознавать лица даже по частичным изображениям. Достижение этого уровня позволит моделям глубокого обучения превзойти способности человека. Описанные далее методы в основном опираются на

известную работу по распознаванию частичных изображений лиц на основе глубокого обучения (Elmahmudi, Ugail, 2019).

Основной подход здесь заключается в использовании признаков, извлеченных из предварительно обученной модели, и применении стандартного классификатора, чтобы увидеть, как различные части лица (рис. 6.10) внедряются в модель на этапе обучения. Затем обученную модель можно использовать для распознавания лиц и решения задачи сопоставления. В данном конкретном случае для обучения и извлечения признаков мы использовали стандартную модель VGG-F. Различные части лица, рассматриваемые здесь, включают глаза, нос, рот, верхнюю и нижнюю половины лица, левую половину лица, 3/4 лица и полное лицо. Далее все извлеченные признаки из модели VGG-F могут быть переданы обоим классификаторам, в данном случае SVM и CS. Лица в VGG-F могут быть представлены с различными частями лица (W) или без них (Wo), чтобы увидеть различия в результатах.

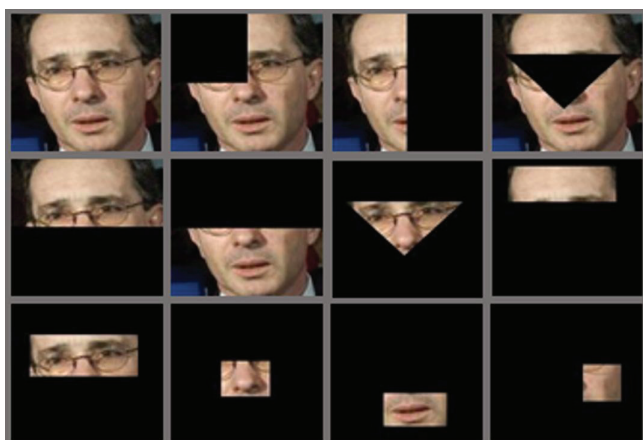


Рис. 6.10 ❖ Пример неполных изображений лиц из набора данных LFW. Для экспериментов по частичному распознаванию лица можно рассматривать половину лица, 3/4 лица и ключевые части лица, такие как глаза, нос, рот и лоб

Например, классификационная способность SVM и CS может быть проверена без использования частей лиц в процессе обучения (SVM-Wo и CS-Wo) и с частями (SVM-W и CS-W). Чтобы исследовать показатели распознавания для каждой части лица, классификаторы можно применять по отдельности. В случае обучения без частей лица видно, что в целом CS-Wo превосходит SVM-Wo для большинства областей лица. Результаты этих экспериментов представлены на рис. 6.11 и 6.12. Видно, что показатели распознавания правой щеки, рта, лба и носа низкие, около 1 % для обоих классификаторов. Напротив, скорость распознавания значительно увеличивается для частей лица, таких как глаза, и достигает 40 % при использовании CS-Wo. Мы также заметили, что по мере увеличения видимой части лица точность распознавания также значительно улучшалась, при этом наилучшая точность распознавания составляла почти 100 % для 3/4 лица и полного изображения

лица. Также следует отметить, что для всех тестов, проведенных в этих экспериментах, показатели CS оказались лучше, чем показатели SVM.

Из результатов, представленных на рис. 6.11 и 6.12 для экспериментов с частичным изображением лица с использованием набора данных FEI, самая высокая точность распознавания (что касается части лица) отмечена для изображений 3/4 лица с использованием SVM-Wo. В этих экспериментальных условиях обучающая выборка не содержала различные части лица. Кроме того, в случае CS-Wo правая половина лица, верхняя половина и 3/4 лица обеспечивают высокие показатели распознавания. Наихудшие показатели распознавания относятся к меньшим по размеру и, возможно, менее значимым частям лица, таким как щека, рот и нос. При применении той же методики к набору данных LFW и обучении с большими пропорциями лица наблюдается небольшое снижение показателей распознавания по сравнению с набором данных FEI, которое составляло от 76 до 99 % для SVM-Wo и от 83 до 99 % для классификатора CS-W. Согласно результатам, полученным для меньших областей лица, наихудшая точность распознавания наблюдается для щек, рта, лба и носа. Однако, судя по всему, глаза содержат больше информации.

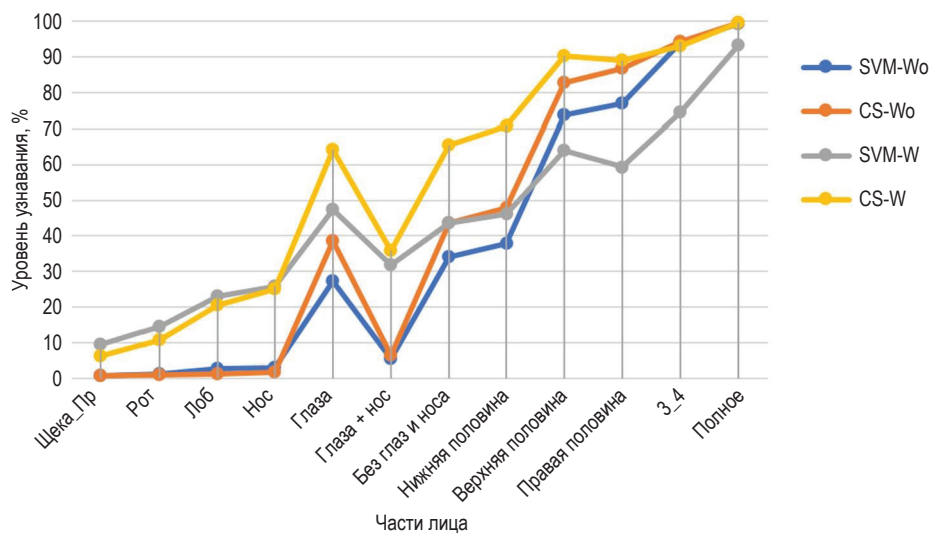


Рис. 6.11 ❖ Результаты распознавания лиц по фрагментам с использованием признаков VGG-F. Точность распознавания (%) измерена с использованием изображений из набора данных LFW с применением классификаторов SVM и CS

Когда к обучающим наборам добавляются отдельные части лица, наблюдается резкое улучшение качества распознавания. Например, точность распознавания правой щеки улучшилась с 0 до 15 % при использовании набора данных FEI. Также следует отметить, что при использовании наборов данных FEI и LFW глаза по-прежнему дают самый высокий уровень распознавания по сравнению с другими частями лица, при этом объединение признаков глаз и носа обеспечивает точность около 90 % при использовании контролируе-

мого набора данных FEI. Однако в случае набора данных LFW этот показатель несколько меньше. Кроме того, мы заметили, что в целом лучшие результаты распознавания достигаются при использовании меры CS.

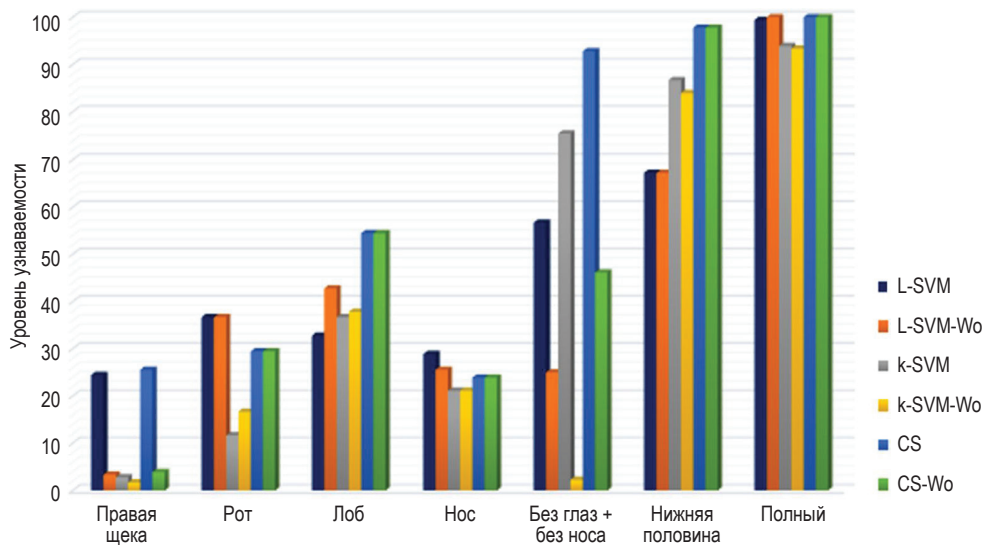


Рис. 6.12 ❖ Результаты распознавания лиц по фрагментам с использованием признаков VGG-F. Результаты распознавания (%) при удалении некоторых частей лица (щеки и часть лица без глаз и носа) из обучающих наборов. Были протестированы следующие классификаторы: линейный SVM, ядерный SVM и CS

Следовательно, в данном случае важно подчеркнуть, что мера CS в целом выглядит лучшим классификатором по сравнению как с линейной, так и с нелинейной SVM. Мера SVM требует полного переобучения при добавлении новых данных, что впоследствии приводит к вычислительным проблемам. Однако в случае классификатора CS таких проблем нет. Хотя на этапе тестирования классификатор CS более требователен к вычислительным ресурсам, но с учетом большей точности имеет смысл использовать классификатор CS вместо SVM.

6.5. ОБУЧЕНИЕ СПЕЦИАЛЬНОЙ МОДЕЛИ ДЛЯ ПОЛНЫХ И ЧАСТИЧНЫХ ИЗОБРАЖЕНИЙ ЛИЦА

В этом разделе мы обсудим, как можно обучить определенные модели глубокого обучения эффективно распознавать лица с использованием полных или частичных изображений. Мы покажем, как можно построить экспериментальный фреймворк для обучения конкретных CNN, настроенных на определенные части лица. При проектировании систем такого рода важно

учитывать уровень используемых обучающих данных, сложность модели и продолжительность обучения, необходимого для создания данной модели.

Предположим, мы создаем модель глубокого обучения, которая точно настроена для идентификации людей только по изображениям их глаз. Начнем с базовой архитектуры VGG-16 (Parkhi et al., 2015), которая аналогична обсуждавшейся ранее VGG-F. Эта модель состоит из 13 сверточных слоев (CONV) с одинаковыми фильтрами размера 3×3 . Сами эти слои делятся следующим образом. Первые два слоя имеют глубину 64. Есть два слоя с глубиной 128, три слоя с глубиной 256 и шесть слоев с глубиной 512. За этими слоями следуют три слоя FC, из которых два слоя содержат 4096 нейронов, а последний слой имеет 1000 нейронов. Структура VGG-16 считается особенно удачной, поскольку она может эффективно управлять количеством гиперпараметров. В частности, расположение слоев было тщательно продумано, чтобы свести к минимуму количество гиперпараметров. Это расположение структурировано следующим образом. Размер фильтров CONV составляет 3×3 , и есть маски 2×2 для слоев заполнения и max-пулинга со страйдом 2. Это говорит о том, что для обучения модели распознаванию определенной части лица мы должны использовать двухэтапную стратегию. На первом этапе первая модель будет построена и обучена только на наборе данных изображений глаз. На втором этапе сгенерированная модель будет использована для построения комплексной модели с использованием той же структуры, что и сеть VGG, с применением тонкой настройки (Yosinski et al., 2014). На рис. 6.13 показана блок-схема процесса обучения модели.

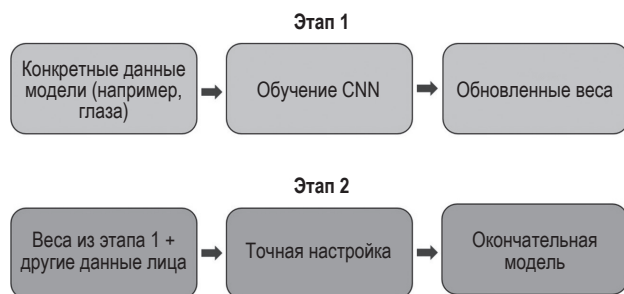


Рис. 6.13 ❖ Процедура обучения модели для распознавания по конкретным частям лица. Процесс обучения может состоять из двух этапов, на которых генерируются веса для определенной части лица, применяемые впоследствии для создания окончательной формы модели

Процесс тонкой настройки алгоритмов машинного обучения – это процедура, при которой выбранная модель CNN, уже обученная для данной работы и/или типа данных, используется для выполнения аналогичной задачи. Это делается путем замены выходного слоя, который изначально был обучен распознаванию предыдущих классов, на слой, который может распознавать новые классы для другой задачи. Преимуществом использования тонкой настройки является сокращение времени вычислений на этапе обучения. На самом деле первые слои уже способны справиться с новой задачей, а обуче-

нию будут подвергнуты только последние слои без необходимости обучения всей сети с нуля. Еще одним преимуществом является повышение точности, поскольку исходные модели обычно обучаются на больших наборах данных.

6.5.1. Предлагаемая архитектура модели

Как было сказано выше, предложенная нами модель для обучения распознаванию по определенным частям лица основана на модели VGG-16 (Parkhi et al., 2015) и требует входного изображения фиксированного размера (224, 224, 3). Модель имеет пять сверточных блоков, каждый из которых содержит ряд сверточных слоев, а за ними следуют нелинейные функции активации, как показано на рис. 6.14.

Как видно по рис. 6.14, предложенная модель CNN для распознавания по частям лица архитектурно похожа на модель VVG-F с пятью сверточными слоями и тремя слоями FC. Эти сверточные слои предназначены для интеграции различных слоев с максимальным объединением для создания эффективных карт объектов. Важнейшим аспектом предлагаемого расположения сверточных и полносвязных слоев является сокращение количества обучаемых параметров (как определено уравнением 6.1), чтобы сохранить разумную продолжительность обучения при сохранении точности сети в допустимых пределах.

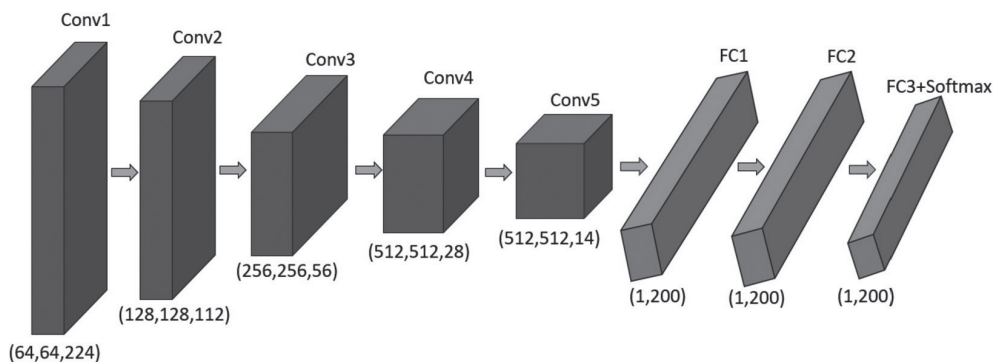


Рис. 6.14 ❖ Архитектура модели CNN для распознавания лиц по отдельным частям. Эта модель является производной от стандартной VGG-16

6.5.2. Фаза обучения модели

Обучение модели CNN – это процедура, позволяющая свести к минимуму различия между эталонными метками и прогнозируемыми выходными данными из набора обучающих данных. Это достигается за счет размещения обучаемых параметров (ядер и весов) в сверточных и полносвязных слоях. В предложенной здесь модели веса фильтров инициализируются с использованием гауссова и стандартного отклонений (Bishop, 2006), а смещения обнуляются. Чтобы оценить точность модели посредством прямого распростра-

нения, можно использовать наиболее часто применяемую функцию потерь для задач множественной классификации, называемую *кросс-энтропией*, или *перекрестной энтропией* (Busoni et al., 2011). Значение функции потерь определяет, насколько хорошо или плохо работает модель после каждой итерации оптимизации. Кроме того, на основе градиента ошибки в текущем состоянии модели все обучаемые параметры будут обновляться во время оптимизации, например с использованием градиентного спуска в сочетании с обратным распространением. Скорость обучения – это еще один гиперпараметр, от которого зависит, насколько быстро модель CNN обучается, используя представленные ей данные. Значение этого параметра положительное и находится в диапазоне от 0 до 1.

Последние два гиперпараметра модели означают количество эпох обучения, через которые должна пройти модель, т. е. сколько раз метод обучения просматривает набор обучающих данных. Необходимо также указать *размер пакета* (batch size, размер выборки обучающих данных, размер партии), необходимый для выполнения эпохи, – рекомендуемый размер пакета составляет 64. Что касается непосредственно обучения, полезно следовать двухэтапному процессу. На первом этапе модель обучается с использованием около 20 эпох с размером пакета 64. После этого сохраняются полученные веса из модели. Эти веса затем используются для инициализации весов на втором этапе обучения, который состоит из 50 эпох. Эпохи разделены на десять частей по пять эпох в каждой. Таким образом, веса из предыдущего этапа обучения используются для инициализации новых весов и обучения модели в течение пяти эпох. Полученные новые веса будут использоваться для следующего тренировочного прогона. Эта процедура продолжается до тех пор, пока не будет достигнут заданный уровень потерь при обучении и тестировании. Во время процесса необходимо стремиться к небольшим потерям при обучении (т. е. приблизительно $< 0,02$) и более высокой точности тестирования (т. е. $> 85\%$).

Что касается обучающих данных, должно быть доступно достаточное количество изображений лиц и соответствующих им персон. Типичные значения – 70 000 изображений, соответствующих определенной части лица у 200 персон. Такой набор данных можно разделить на две группы: 70 % для обучения и 30 % для проверки. Наконец, обучающий набор используется для обучения модели, а потери вычисляются путем прямого распространения, тогда как обновление обучаемых параметров происходит посредством обратного распространения.

6.6. ЗАКЛЮЧЕНИЕ

В этой главе мы обсудили современное состояние дел в области распознавания лиц с использованием методов и приемов глубокого обучения. Мы показали, как можно использовать методы глубокого обучения для сопоставления личностей и выявления сходства лиц – с использованием как полных изображений лиц в анфас, так и частичных изображений лица. Мы показали,

как готовые признаки хорошо обученных моделей можно использовать для создания эффективных и точных систем распознавания лиц. Мы также показали, как обучать определенные модели, построенные на основе различных архитектур, обучаемых слоев и весов, заимствованных из хорошо известных глубоких моделей, которые используются для обработки и анализа изображений.

Например, мы исследовали вопрос, связанный с идеей распознавания лиц по характерным фрагментам. В частности, показали, как работает глубокое распознавание лиц, когда такие части, как глаза, рот, нос и щека, используются в качестве источника признаков для обучения и распознавания. Мы также показали, что системы распознавания лиц могут использовать несколько подходов. Например, продемонстрировали, как реализовать современную CNN, инкапсулирующую в себя предварительно обученные модели (такие как VGG-F), с помощью которых можно извлечь ключевые признаки лица. Затем можно использовать хорошо известные классификаторы, такие как мера косинусного подобия и машины опорных векторов, для проверки точности распознавания. Точно так же можно применять обучение моделей на конкретных частях лица, таких как рот, нос, глаза, лоб и их комбинации, для разработки эффективных систем распознавания лиц по частичным изображениям.

Очевидно, что развитие методов и техник глубокого обучения принесло огромную пользу отрасли распознавания лиц (Guo and Zhang, 2019). Многие проблемы, ранее считавшиеся неразрешимыми, теперь считаются простыми. Например, при наличии двух хорошо освещенных фронтальных изображений человека подтверждение совпадения личности с характерных признаков лица теперь считается тривиальной задачей. Однако в целом распознавание лиц по-прежнему остается актуальным направлением исследований, в котором остается ряд сложных и нерешенных проблем. Например, проблема компьютерного распознавания лиц с использованием частичных данных о лице до сих пор остается в значительной степени неисследованной областью. Учитывая, что люди и компьютеры выполняют распознавание лиц и аутентификацию по-разному, интересно понять, как компьютер будет реагировать на различные части лица, когда они предъявляются в задаче распознавания лиц.

Компьютерное распознавание лиц, начиная с 1960-х годов, прошло долгий путь. Тем не менее осталось много проблем и препятствий, которые еще предстоит преодолеть. Это, например, распознавание лиц в следующих проблемных ситуациях: плохое освещение, разные положения, неполное изображение, перевернутые лица, постаревшие лица и изображения, полученные с больших расстояний. Вопрос о точности измерения сходства лиц с использованием методов глубокого обучения в значительной степени остается без ответа. Например, на вопрос о сходстве между лицами, связанными близкими родственными отношениями, – братьями, сестрами и однойцевыми близнецами – пока нет удовлетворительного ответа. Кроме того, нет удовлетворительных ответов на вопросы о предвзятости обучающих данных и о том, как сделать системы глубокого обучения прозрачными и объяснимыми.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Bishop C. M., 2006. Pattern Recognition and Machine Learning. Springer, Berlin.
- Busoni L., Ernst D., De Schutter B., Babuska R., 2011. Cross-entropy optimization of control policies with adaptive basis functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 41 (1), 196–209.
- Chatfield K., Simonyan K., Vedaldi A., Zisserman A., 2016. Return of the devil in the details: delving deep into convolutional nets. In: *Proceedings of the British Machine Vision Conference (BMVC)*.
- Day O., Khoshgoftaar T. M., 2017. A survey on heterogeneous transfer learning. *Journal of Big Data* 4 (1).
- Dora L., Agrawal S., Panda R., Abraham A., 2017. An evolutionary single Gabor kernel based filter approach to face recognition. *Engineering Applications of Artificial Intelligence* 62, 286–301.
- Dumoulin V., Visin F., 2018. A guide to convolution arithmetic for deep learning. *arXiv:1603.07285v2*.
- Elmahmudi A., Ugail H., 2019a. The biharmonic eigenface. *Signal, Image and Video Processing* 3, 1639–1647.
- Elmahmudi A., Ugail H., 2019b. Deep face recognition using imperfect facial data. *Future Generations Computer Systems* 41 (99), 213–225.
- Gholamalinezhad H., Khosravi H., 2020. Pooling methods in deep neural networks, a review. *arXiv:2009.07485*.
- Guo G., Zhang N., 2019. A survey on deep learning based face recognition. *Computer Vision and Image Understanding* 189, 102905.
- He K., Zhang X., Ren S., Sun J., 2015. Deep residual learning for image recognition. In: *Proceedings of the British Machine Vision Conference (BMVC)*.
- He L., Li H., Zhang Q., Sun Z., 2018. Dynamic feature learning for partial face recognition. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7054–7063.
- Huang G. B., Ramesh M., Berg T., Learned-miller E., 2008. Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, pp. 1–11.
- Jolliffe I. T., 2002. *Principal Component Analysis*. Springer, New York.
- Kas M., Elmerabet Y., Ruichek Y., Messoussi R., 2020. A comprehensive comparative study of handcrafted methods for face recognition LBP-like and non LBP operators. *Multimedia Tools and Applications* 79, 375–413.
- Krizhevsky A., Sutskever I., Hinton G. E., 2012. ImageNet classification with deep convolutional neural networks. In: *NIPS*.
- LeCun Y., Bengio Y., Hinton G., 2015. Deep learning. *Nature* 521, 436–444.
- Liu W., Wang Z., Liu X., Zeng N., Liu Y., Alsaadi F. E., 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234, 11–26.
- Parkhi O. M., Vedaldi A., Zisserman A., 2015. Deep face recognition. In: *IEEE CVPR*.
- Schroff F., Kalenichenko D., Philbin J., 2015. FaceNet: A unified embedding for face recognition and clustering. In: *IEEE CVPR*.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V.,

- Rabinovich A.*, 2015. Going deeper with convolutions. In: IEEE CVPR.
- Taigman Y., Yang M., Ranzato M., Wolf L.*, 2014. DeepFace: closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 701–1708.
- Thomaz C. E., Giraldi G. A.*, 2010. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing* 28 (6), 902–913.
- Turk M., Pentland A.*, 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 13, 71–86.
- Weiss K., Khoshgoftaar T. M., Wang D.*, 2016. A survey of transfer learning. *Journal of Big Data* 3 (1).
- Yosinski J., Clune J., Bengio Y., Lipson H.*, 2014. How transferable are features in deep neural networks? In: *Proceedings of the Advances in Neural Information Processing Systems*.
- Young A. W., Burton A. M.*, 2018. Are we face experts. *Trends in Cognitive Sciences* 22, 100–110.
- Zeiler M. D., Fergus R.*, 2014. Visualizing and understanding convolutional networks. In: *Computer Vision – ECCV 2014*.

ОБ АВТОРЕ ГЛАВЫ

Хассан Угайл – директор Центра визуальных вычислений факультета инженерии и информатики Университета Брэдфорда, Великобритания. Он имеет степень бакалавра с отличием по математике в Королевском колледже Лондона и докторскую степень в области начертательной геометрии в Школе математики при Университете Лидса. Научные интересы профессора Угайла включают компьютерное геометрическое и функциональное проектирование, визуализацию и машинное обучение.

Глава 7

Адаптация домена с использованием неглубоких и глубоких нейросетей, обучаемых без учителя

Авторы главы:

Йогеш Балахи, факультет компьютерных наук и UMACS,
Мэрилендский университет, Колледж-Парк, Мэриленд, США;
Хиен Нгуен, факультет электроники и вычислительной техники,
Хьюстонский университет, Хьюстон, Техас, США;
Рама Челлаппа, факультеты электроники,
вычислительной техники и биомедицинской инженерии,
Университет Джона Хопкинса, Балтимор, Мэриленд, США¹

Краткое содержание главы:

- адаптация домена с использованием неглубоких¹ и глубоких нейросетей;
- интерполяция между исходным и целевым доменами с использованием многообразий и словарей;
- генеративные состязательные сети как посредники при переходе между доменами.

7.1. ВВЕДЕНИЕ

Располагая обширными данными, полученными с различных устройств и при разных условиях, мы часто оказываемся в ситуации, когда данные,

¹ Неглубокие нейросети (shallow neural network) – это сети, состоящие из одного или двух слоев. – *Прим. перев.*

используемые для обучения классификатора, по некоторым показателям отличаются от тех, что представлены во время прикладного использования. Подобные случаи нередко возникают в приложениях для распознавания объектов, где обучающие и рабочие данные записываются при различных условиях освещения; при обработке речи, когда разговорная модель, обученная на студийных записях, должна быть развернута в более реалистичной внешней среде; при индексации файлов мультимедиа, когда легко доступны размеченные фотографии Flickr или видео YouTube, на основе которых пользователь хотел бы автоматически проиндексировать свою собственную фото- и видеокolleкцию, собранную с помощью бытовой камеры. *Адаптация домена*¹ (domain adaptation, DA) относится к классу методов, направленных на изучение представлений из набора данных с небольшими ресурсами (объем, разметка) путем передачи знаний из связанного, но отличающегося набора данных, имеющего богатые ресурсы. Различия между наборами данных могут выражаться в разнице освещения, разных узорах текстур или любых подобных расхождениях, присущих набору данных. Для передачи имеющихся знаний в другой домен обычно ищут представление признаков, не зависящее от домена, которое эффективно устраняет мешающие расхождения.

Обучение с переносом (transfer learning, TL) (Pan et al., 2010) – это родственный подход, который решает проблему расхождений между наборами данных. Основное различие между TL и DA связано с тем, какие свойства данных сохраняются в условиях обучения и применения. В то время как TL имеет дело со случаем, когда условное распределение меток данных меняется (т. е. задачи в двух доменах различны), а предельное распределение данных сохраняется, DA обращается к противоположному сценарию (Daume and Marcu, 2006), где распределение данных между двумя доменами различается, но задача остается прежней. Хотя нам часто приходилось видеть, как практикующие специалисты взаимозаменяемо применяют методы TL и DA в том и другом случае, в этой главе мы сосредоточимся на DA, поскольку данный метод естественным образом подходит к приложениям компьютерного зрения, таким как распознавание объектов, когда пользователь заинтересован в сохранении идентичности разных вариаций одного объекта, независимо от ракурса и освещения.

Существуют две широкие категории методов DA в зависимости от того, имеют ли тестовые данные целевого домена неполную разметку (обучение с частичным участием учителя) или совсем не имеют разметки (обучение без учителя). В то время как метод DA с частичным участием учителя нередко использует соответствие из размеченных целевых данных для изучения перехода к целевому домену (Daume, Marcu, 2006; Saenko et al., 2010), DA без учителя использует стратегии, которые предполагают (а) определенный класс преобразований между доменами (Wang, Mahadevan, 2009), (б) наличие отличительных признаков, которые являются общими или инвариантными для обоих доменов (так называемые «инварианты доменов») (Blitzer et al., 2008; Mansour et al., 2009), или (в) скрытое пространство, где разница в рас-

¹ Напомним, что доменом мы для краткости называем предметную (тематическую) область, которую охватывает модель. – *Прим. перев.*

пределении исходных и целевых данных минимальна (Blitzer et al., 2011). Помимо адаптации между одним исходным и одним целевым доменами, были проведены исследования многодоменной адаптации (например, Mansour et al., 2009), в которых рассматривалась ситуация с наличием более одного исходного и/или целевого домена. В то время как некоторые из этих подходов преследуют цель «адаптации представления» путем изучения преобразования различия между доменами, другие (Duan et al., 2009; 2012) выступают за подход, ориентированный на «классификатор», который пытается получить целевые классификаторы путем манипулирования или повторной оптимизации классификаторов, обученных на исходном домене. В оставшейся части главы мы сосредоточимся в основном на обучении без учителя.

С 2011 г. было проведено значительное количество исследований, направленных на решение задачи адаптации домена с обучением без учителя. К ним относятся методы, основанные на дифференциальной геометрии (Gong et al., 2011; 2014; Gong et al., 2012; Ho, Gopalan, 2014), разреженных словарях (Lu et al., 2015; Nguyen et al., 2012; 2015; Shekhar et al., 2013; Xu et al., 2015), а относительно недавно были разработаны генеративно-состязательные сети (GAN) (Sankaranarayanan et al., 2018; Shi and Sha, 2012). В этой главе мы рассмотрим несколько типичных примеров из трех упомянутых выше подходов. За более подробным изложением исследований в области адаптации доменов читатель может обратиться к работе (Patel et al., 2015).

7.2. АДАПТАЦИЯ ДОМЕНА С ИСПОЛЬЗОВАНИЕМ МНОГООБРАЗИЯ

Некоторые исследователи, действуя в духе «постепенного перехода между крайностями» (Gopalan et al., 2014), предложили вариант DA с обучением без учителя, в основе которого лежит построение гладкого пути между исходным и целевым доменами с использованием промежуточных представлений данных, которые передают соответствующую информацию о расхождении между доменами. Не делая предположений об инвариантных свойствах домена, авторы этого метода представили первый метод DA с обучением без учителя для распознавания объектов путем вычисления пути адаптации определенных статистических характеристик от исходного домена (доменов) к целевому домену (доменам). Частный случай этой схемы, представленный в (Gopalan et al., 2011), использует в качестве представления домена линейное порождающее подпространство. Точнее, к каждой из областей применяется анализ главных компонент с последующим представлением подпространств в виде точек на многообразии Грассмана. Затем геодезическая линия между этими точками используется как статистически значимый путь для представления различий между доменами. Путем выборки точек вдоль геодезической линии получают промежуточные кросс-доменные представления данных, с помощью которых дискриминативный классификатор обучается выполнять распознавание. В этой работе впоследствии были

рассмотрены другие частные случаи этой стратегии, такие как представление домена в многомерном *гильбертовом пространстве воспроизводящего ядра* (reproducing kernel Hilbert space, RKHS) с использованием методов ядра и представление многообразия низкой размерности с использованием *собственных отображений матрицы Лапласа* (Laplacian Eigenmaps). Интересно, что эта структура также поддерживает варианты адаптации с частичным участием учителя и несколькими доменами и была дополнительно улучшена за счет моделирования мелкомодульных доменов в сочетании с постепенно меняющимися пропорциями исходных и целевых экземпляров, а также за счет применения бустинга к пулу промежуточных представлений, полученных по разным параметрам.

С момента публикации (Gopalan et al., 2011), были проведены и другие схожие исследования, такие как (Gong et al., 2012 г.; Zheng et al., 2012), в которых обсуждались альтернативные стратегии выборки вдоль геодезической линии, и (Shi, Sha, 2012), которые предложили теоретико-информационный подход для совместного изучения признаков сдвига предметной области и классификаторов. Адаптация с несколькими источниками, которая может учитывать различные типы объектов в разных доменах за счет использования механизма регуляризации, зависящего от данных, была рассмотрена в работе (Duan et al., 2012), а устойчивость к шуму или выбросам была разобрана в методе реконструкции низкого ранга (Jhuo et al., 2012). Дополнительный анализ и сравнительные оценки многих из этих значимых исследований обобщены в (Patel et al., 2015).

7.2.1. Адаптация домена без учителя с использованием произведения многообразий

Применение *адаптации домена без привлечения учителя* (unsupervised domain adaptation, UDA) для неограниченного распознавания лиц представляет собой очень сложную проблему из-за различий внешнего вида лица на разных изображениях, вызванных множеством факторов, таких как размытие, выражение, освещение, ракурс и разрешение. В результате классификаторы лиц, обученные с предположением, что обучающие и тестовые данные взяты из схожих распределений, обычно имеют очень низкую точность, особенно при применении в средах без учителя. Например, алгоритмы распознавания лиц, обученные на образцах из исходного домена, содержащего четкие, хорошо освещенные изображения лиц, плохо работают при использовании в целевом домене, содержащем размытые, плохо освещенные изображения лиц (Vageeswaran et al., 2013). Качество работы этих алгоритмов еще больше ухудшается, когда доступно только ограниченное количество изображений каждого лица из-за высокой стоимости и других проблем при сборе данных.

Несмотря на то что было проведено несколько исследований, посвященных заранее заданным вариациям лица в исходном и целевом доменах (Zhao et al., 2003), таких как исследование девятиточечного источника освещения

(Lee et al., 2005), анализу различий между доменами, вызванных множественными неизвестными факторами, не уделялось должного внимания. Адаптация доменов – это новая парадигма для решения таких преобразований в более широком контексте, в соответствии с которой при наличии помеченных данных из исходного домена и небольшого количества (или отсутствия) помеченных данных из целевого домена разрабатываются подходы обучения без учителя или с частичным привлечением учителя для настройки модели на различия в данных между доменами (Saenko et al., 2010; Ben-David et al., 2010; Gopalan et al., 2011). Большинство этих методов учитывают различия доменов в статистическом смысле, поскольку модели, вызывающие изменения в данных, неизвестны. Это ограничивает их применение конкретной проблемой распознавания лиц, где имеются богатые наработки по моделям ракурса, освещения, размытия, выражения и старения. Важно понимать различия доменов с точки зрения основных ограничений, относящихся к моделям, которые генерируют наблюдаемые данные. Такой анализ требует изучения геометрических свойств пространства изображений, индуцированных этими моделями.

Однако многие традиционные подходы часто либо игнорируют геометрические структуры пространства, либо наивно рассматривают пространство как евклидово (Lui, 2012). Хотя нелинейные алгоритмы изучения многообразия, такие как ISOMAP (Tenenbaum et al., 2000) или локально-линейное представление (locally linear embedding, LLE) (Roweis, Saul, 2000), предлагают альтернативы, они требуют больших объемов обучающих данных для лежащей в основе доменов нелинейной многообразной структуры данных. Подобное требование к данным не всегда может быть удовлетворено во многих реальных приложениях. Одним из возможных способов обработки вариаций лица, возникающих из-за множества факторов, является использование математического инструмента, называемого *полилинейной алгеброй*, т. е. алгеброй тензоров более высокого порядка. Поскольку матрицы представляют линейные операторы над векторным пространством, их обобщение, тензоры, определяют полилинейные операторы над множеством векторных пространств (Vasilescu, Terzopoulos, 2002). Хотя существуют исследования, посвященные использованию полилинейной алгебры для распознавания лиц (Vasilescu and Terzopoulos, 2002; 2007), такие подходы игнорируют искривленную геометрию пространства изображения и оперируют евклидовым пространством. В ряде работ сообщалось о попытках включить нелинейные геометрические структуры в структуру тензорных вычислений (Lui, Beveridge, 2010; Park, Savvides, 2011), но им снова нужны большие обучающие данные.

В работе (Ho, Gopalan, 2014) представлено адаптивное решение для распознавания лиц на основе тензорной геометрии моделей, объясняющих вариации лица всего с одним изображением на человека в исходном домене. Вместо того чтобы находить линейные преобразования, представляющие переход между доменами, как в (Saenko et al., 2010; Kulis et al., 2011), мы предлагаем основанный на модели подход к построению латентного домена, в котором многофакторные вариации лица в исходном и целевом доменах могут быть захвачены вместе. Одним из основных преимуществ такого подхода является то, что даже если данные в пределах исходного домена и/или

целевого домена неоднородны, например когда *расхождение доменов*¹ вызвано размытием, а исходные и целевые данные содержат сочетание четких и размытых лиц, процесс изучения смещения доменов остается неизменным, в отличие от других методов, которые предполагают, что домены будут более или менее однородными (Saenko et al., 2010; Kulis et al., 2011; Gopalan et al., 2011). Кроме того, предлагаемый метод преодолевает ограничение требований к данным для моделирования вариаций домена путем синтеза нескольких изображений лица при различном освещении, размытии и ракурсе из одного входного изображения в исходном или целевом домене и использует их для формирования многомерного тензора, в отличие от других методов, таких как (Lui, Beveridge, 2010), которые предъявляют более строгие требования к данным. Затем тензор, полученный из набора синтезированных изображений, можно представить на произведении многообразий, выполнив *разложение по сингулярным значениям высшего порядка* (higher-order singular value decomposition, HOSVD) и сопоставив каждую ортогональную факторизованную матрицу с точкой на многообразии Грассмана. *Порядок тензоров* – это количество факторов, используемых в процессе синтеза. Далее мы распознаем метки лиц целевого домена, выполняя вычисления, относящиеся к тензорной геометрии, для случаев, когда исходный домен либо содержит только одно изображение для каждого субъекта, либо имеет несколько изображений для каждого субъекта. В этой работе также рассматривается проблема сопоставления наборов изображений, которая связана с распознаванием лиц на основе видео, когда несколько кадров в видео предоставляют доказательства, связанные с идентификацией лица.

Методы, основанные на многообразии, по-видимому, потеряли популярность из-за более высокой точности методов, основанных на генеративно-состязательных сетях.

7.3. АДАПТАЦИЯ ДОМЕНА БЕЗ УЧИТЕЛЯ С ИСПОЛЬЗОВАНИЕМ СЛОВАРЕЙ

Разреженные и избыточные представления сигналов вызвали большой интерес в области компьютерного зрения, обработки сигналов и изображений (Bruckstein et al., 2009; Elad et al., 2010; Rubinstein et al., 2010; Wright et al., 2010; Bo et al., 2011). Отчасти это связано с тем, что интересующие сигналы и изображения могут быть представлены разреженно или сжимаемы при наличии соответствующего словаря. В частности, мы говорим, что сигнал $y \in R^n$ разреженно представлен словарем $D \in R^{n \times K}$, если он может быть хорошо аппроксимирован линейной комбинацией нескольких столбцов D , когда $y \approx Dx$, где $x \in R^K$ – вектор разреженного представления, а D – словарь, кото-

¹ Авторы книги используют термин *domain shift* (смещение домена), но этот термин уже применяется в теории магнетизма для описания явлений, происходящих с магнитными доменами, поэтому мы постараемся избежать путаницы и будем говорить о расхождениях доменов. – Прим. перев.

рый содержит набор базовых элементов (атомов) в качестве столбцов. Поиск разреженного вектора представления влечет за собой решение следующей задачи оптимизации:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|x\|_2 \text{ так, что } \|y - Dx\|_2 \leq \epsilon, \quad (7.1)$$

где ϵ – допустимая ошибка, $\|x\|_0$ – мера нулевой разреженности, которая отражает количество ненулевых элементов в векторе x , а $\|y - Dx\|_2$ – среднеквадратическая ошибка, полученная в результате разреженной аппроксимации. Решение (7.1) является NP-трудным и может быть аппроксимировано различными методами (Chen et al., 2001; Patil et al. 1993; Tropp, 2004). Вместо использования заранее определенного словаря можно напрямую изучить словарь из данных. Действительно, было замечено, что изучение словаря непосредственно из обучающих данных вместо использования заранее определенного словаря (например, вейвлета) обычно приводит к более компактному представлению и, следовательно, может обеспечить лучшие результаты во многих приложениях обработки изображений, таких как восстановление и классификация (Elad et al., 2010; Rubinstein et al., 2010; Wright et al., 2010; Olshausen and Field, 1996; Mairal et al., 2009, 2011). Для задачи изучения словаря было разработано несколько алгоритмов. Двумя наиболее известными алгоритмами являются *метод оптимальных направлений* (method of optimal directions, MOD) (Engan et al., 1999) и алгоритм K-SVD (Aharon et al., 2006). Для заданного множества N сигналов $Y = [y_1, y_2, \dots, y_N]$ цель алгоритмов K-SVD и MOD состоит в том, чтобы найти словарь D и разреженную матрицу X , которые минимизируют следующую ошибку представления:

$$(D, X) = \underset{D, X}{\operatorname{argmin}} \|Y - DX\|_F^2 \text{ так, что } \|x_i\| \leq T_0, \forall i = 1, \dots, N, \quad (7.2)$$

где x_i представляет i -й столбец X , $\|A\|_F$ обозначает фробениусову норму A , а T_0 обозначает уровень разреженности. И MOD, и K-SVD являются итеративными методами, которые чередуют этапы разреженного кодирования и обновления словаря. Сначала инициализируется словарь D с ℓ_2 -нормализованными столбцами. Следующая за этим основная итерация состоит из двух этапов:

- **разреженное кодирование:** на этом шаге D не меняется и решается следующая задача оптимизации для вычисления вектора представления x_i для каждого изображения y_i :

$$\min_{x_i} \|y_i - Dx_i\|_2^2 \text{ так, что } \|x_i\|_0 \leq T_0, \forall i = 1, \dots, N; \quad (7.3)$$

- **обновление словаря:** на этом этапе алгоритмы MOD и KSVD различаются. Алгоритм MOD обновляет все атомы одновременно, решая задачу оптимизации, решение которой дается как $D = YX^+$, где X^+ обозначает псевдоинверсию Мура–Пенроуза. Несмотря на то что алгоритм MOD очень эффективен и обычно сходится за несколько итераций, он страдает от высокой сложности обращения матрицы, как отмечено в (Aharon et al., 2006). В случае K-SVD обновление словаря выполня-

ется эффективным способом, атом за атомом, а не с использованием матричной инверсии. Было замечено, что для сходимости алгоритма K-SVD требуется меньше итераций, чем для метода MOD.

7.3.1. Общий словарь доменной адаптации

Когда распределения целевых и исходных данных различаются, изученное разреженное представление может оказаться неоптимальным. В этом разделе мы рассматриваем возможность оптимального представления источника и цели с помощью общего словаря. В частности, описываем метод, который совместно изучает проекции данных в двух доменах, и скрытый словарь, который может кратко представлять оба домена в спроецированном низкоразмерном пространстве, как показано на рис. 7.1. Данный эффективный метод оптимизации можно легко распространить на несколько доменов. Затем изученный словарь можно использовать для классификации. Предлагаемый подход не требует явного соответствия между исходным и целевым доменами и показывает хорошие результаты, даже когда в целевом домене доступно лишь несколько меток. Различные эксперименты по распознаванию показывают, что этот метод работает наравне или даже лучше, чем конкурирующие методы.

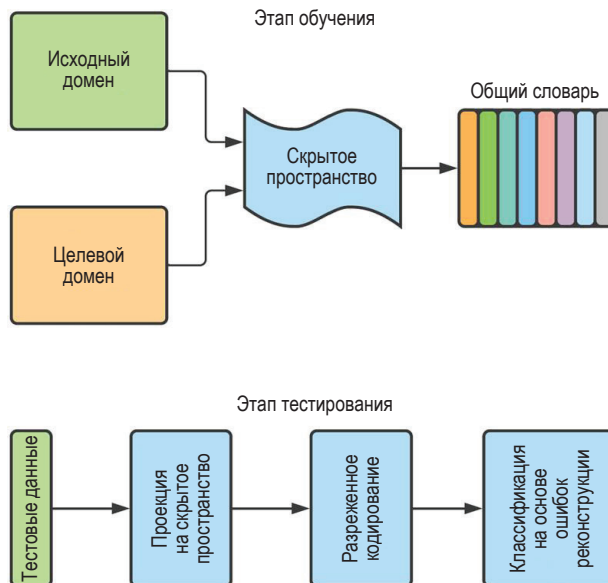


Рис. 7.1 ❖ Схема метода с общим словарем доменной адаптации (Shekhar et al., 2013)

Рассмотрим частный случай, когда у нас есть данные из двух доменов, $Y_1 \in R^{d \times N_1}$ и $Y_2 \in R^{d \times N_2}$. Мы хотим выучить общий словарь K-атомов $D \in R^{n \times K}$ и отображения $P_1 \in R^{n \times d}$, $P_2 \in R^{n \times d}$ на общее пространство малой размерности.

сти, что минимизирует ошибку представления в пространстве проекции. Формально нам нужно минимизировать следующую функцию стоимости:

$$C_1(D, P_1, P_2, X_1, X_2) = \|P_1 X_1 - D X_1\|_F^2 + \|P_2 X_2 - D X_2\|_F^2 \quad (7.4)$$

– с учетом ограничений разреженности на X_1 и X_2 . Далее мы предполагаем, что строки проекционных матриц P_1 и P_2 ортогональны и нормированы к единичной норме. Это предотвращает вырождение решения.

Регуляризация: при переносе данных из двух доменов в общее подпространство преобразования нельзя терять слишком много информации, доступной в исходных доменах. Чтобы облегчить эту задачу, мы добавляем член регуляризации, подобный PCA, сохраняющий энергию в исходном сигнале, заданном как

$$C_2(P_1, P_2) = \|Y_1 - P_1^T P_1 Y_1\|_F^2 + \|Y_2 - P_1^T P_2 Y_2\|_F^2. \quad (7.5)$$

Мы можем записать параметры следующим образом:

$$\tilde{P} = [P_1, P_2], \quad \tilde{Y} = \begin{pmatrix} Y_1 \\ 0 \\ Y_2 \end{pmatrix} \quad \text{и} \quad \tilde{X} = [X_1, X_2]. \quad (7.6)$$

Используя новые обозначения, общую оптимизацию можно переписать следующим образом:

$$\{D^*, \tilde{P}^*, \tilde{X}^*\} = \underset{D, \tilde{P}, \tilde{X}}{\operatorname{argmin}} C_1(D, \tilde{P}, \tilde{X}) + \lambda C_2(\tilde{P})$$

так, что $P_i P_i^T = I, i = 1, 2$ и $\|\tilde{x}_j\|_0 \leq T_0, \forall j$. (7.7)

Поддержка нескольких доменов: приведенную выше формулу можно расширить, чтобы она могла описывать несколько доменов. Для M доменов мы просто строим матрицы $\{\tilde{Y}, \tilde{P}, \tilde{X}\}$ как

$$\tilde{P} = [P_1, P_2, \dots, P_M], \quad \tilde{Y} = \begin{pmatrix} Y_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Y_M \end{pmatrix} \quad \text{и} \quad \tilde{X} = [X_1, X_2, \dots, X_M]. \quad (7.8)$$

Оптимизация: мы минимизируем целевую функцию в уравнении (7.7) путем чередования оптимизации \tilde{P} и (D, \tilde{X}) . В частности, когда фиксируется \tilde{P} , оптимизация становится стандартной задачей изучения словаря, где эффективны алгоритмы K-SVD и MOD. Когда фиксируются (D, \tilde{X}) , мы можем минимизировать уравнение (7.7) с использованием методов оптимизации многообразия (Wen, Yin, 2013). В качестве альтернативы мы можем получить оптимальную форму (P_i, D) и преобразовать целевую функцию в более простую форму перед оптимизацией, как это сделано в работе (Shekhar et al., 2013).

Классификация: как только общий словарь для нескольких доменов изучен, наш метод будет выполнять классификацию с использованием следующей процедуры. Во-первых, мы проецируем тестовую выборку в скрытое

пространство, используя изученное преобразование P_i , выполняем разреженное кодирование, затем вычисляем ошибку реконструкции для каждого преобразования домена. Класс, соответствующий наименьшей ошибке, будет окончательным предсказанием класса. Процедуру можно записать следующим образом:

$$\hat{x}_{te}^i = \underset{x}{\operatorname{argmin}} \|P_i y - Dx\|_F^2 \text{ так, что } \|x\|_0 \leq T_0, \leftrightarrow i; \quad (7.9)$$

$$\text{Выходной класс} = \underset{i=1, \dots, C}{\operatorname{argmin}} \|P_i y - D\hat{x}_{te}^i\|_2. \quad (7.10)$$

Член ошибки в уравнении (7.10) называется *остаточной ошибкой*, которая является мерой несоответствия тестовой выборки определенному классу.

Эксперименты: мы используем набор данных *Multipie* (Gross et al., 2010) – полный набор данных о лицах 337 субъектов, включающий изображения, сделанные в 15 ракурсах, 20 вариантах освещения, при 6 выражениях лица и в течение 4 разных сеансов. В целях нашего эксперимента мы использовали 129 субъектов, общих для сеансов 1 и 2. Эксперимент проводился с использованием 5 ракурсов, от фронтального (анфас) до 75° . Наборы фронтальных снимков лица были взяты в качестве исходного домена, в то время как различные нефронтальные ракурсы были взяты в качестве целевых доменов. Словари обучались с использованием вариантов освещения $\{1, 4, 7, 12, 17\}$ относительно исходного и целевого ракурсов в сеансе 1 для каждого субъекта. Все изображения с разными вариантами освещения из сеанса 2 для целевого ракурса были взяты в качестве тестовых изображений.

Сначала мы рассмотрим задачу *выравнивания ракурса* (pose alignment) с использованием предложенной схемы изучения словаря. Выравнивание ракурса затруднено из-за очень нелинейных искажений, вызванных трехмерным вращением лица. В качестве целевого ракурса были приняты изображения, полученные в крайнем ракурсе 60° . Общий дискриминативный словарь был изучен с использованием подхода, описанного в этом разделе. Конкретное тестовое изображение проецировалось на скрытое подпространство и реконструировалось с использованием словаря. Реконструкция проецировалась обратно на исходный домен ракурса, чтобы получить выровненное изображение. На рис. 7.2(а) показаны синтезированные изображения для различных условий. Из этого рисунка мы можем сделать некоторые полезные выводы о методе. Во-первых, видно, что существует оптимальный размер словаря $K = 5$, при котором достигается наилучшее выравнивание. Кроме того, при изучении дискриминационного словаря сохраняется идентичность субъекта. При $K = 7$ выравнивание не очень хорошее, так как изученный словарь не может успешно сопоставить два домена, когда в словаре больше атомов. Словарь с $K = 3$ дает более высокую ошибку реконструкции, поэтому результат не является оптимальным. Мы выбрали $K = 5$ для дополнительных экспериментов с зашумленными изображениями. Из строк 2 и 3 видно, что предложенный метод устойчив даже при высоких уровнях шума и отсутствующих пикселях. Отметим, что синтезированные изображения были созданы с шумоподавлением и ретушированием дефектов, как показано в строках 2

и 3 на рис. 7.2 (а) соответственно. Это показывает эффективность нашего метода. Изученные матрицы проекций (рис. 7.2b) показывают, что наш метод может изучить внутреннюю структуру двух доменов. В результате он способен выучить надежный общий словарь.

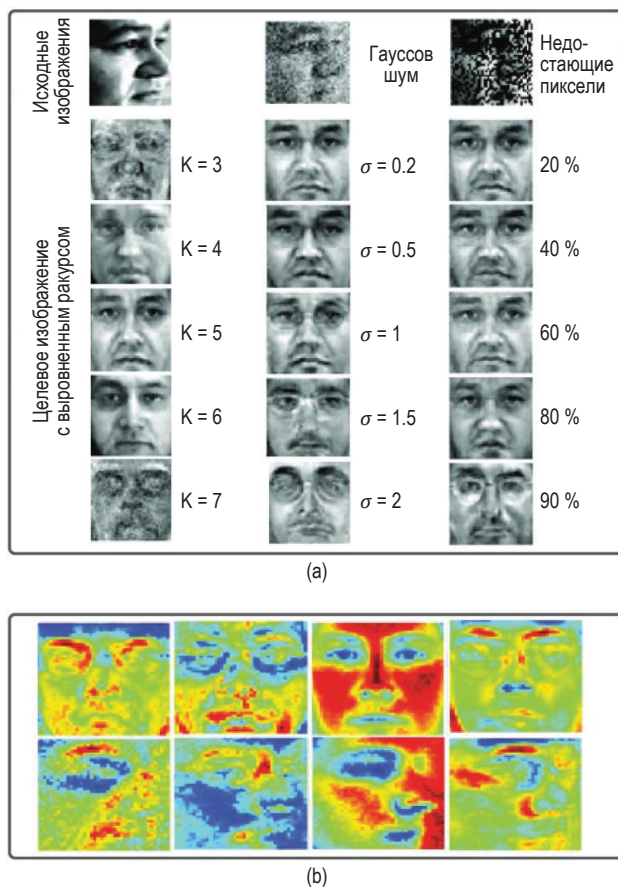


Рис. 7.2 ❖ Результаты выравнивания ракурса (Shekhar et al., 2013). (а) Примеры изображений с выравниванием ракурса с использованием предложенного метода. Синтез в различных условиях демонстрирует надежность метода. (б) Первые несколько компонентов изученных матриц проекций для двух ракурсов

Мы также провели эксперимент по распознаванию, используя описанный выше набор условий. В табл. 7.1 показано, что наш метод выгодно отличается от других алгоритмов распознавания с несколькими представлениями (Sharma et al., 2012) и в среднем дает наилучшую точность. Алгоритм изучения словаря FDDL (Янг и др., 2011) здесь не оптимален, поскольку он не может эффективно представить нелинейные искажения, вызванные изменением ракурса.

Таблица 7.1. Сравнение точности предложенного метода с другими алгоритмами распознавания лиц в разных ракурсах (Shekhar et al., 2013)

Метод	Ракурс					
	15°	30°	45°	60°	75°	Среднее
PCA	15,3	5,3	6,5	3,6	2,6	6,7
PLS (Sharma, Jacobs, 2011)	39,3	40,5	41,6	41,1	38,7	40,2
LDA	98,0	94,2	91,7	84,9	79,0	89,5
CCA (Sharma, Jacobs, 2011)	92,1	89,7	88,0	86,1	83,0	83,5
GMLDA (Sharma et al., 2012)	99,7	99,2	98,6	94,9	95,4	97,6
FDDL (Yang et al., 2011)	96,8	90,6	94,4	91,4	90,5	92,7
SDDL (Shekhar et al., 2013)	98,4	98,2	98,9	99,1	98,8	98,7

7.3.2. Совместная иерархическая адаптация домена и изучение признаков

Сложные визуальные данные содержат отличительные структуры, которые трудно полностью охватить каким-либо одним дескриптором признаков. В то время как описанный выше метод адаптации домена сосредоточен на адаптации одного признака, созданного вручную, важно уметь выполнять адаптацию иерархии признаков, чтобы использовать богатство визуальных данных. В этом разделе обсуждается метод *адаптации домена на основе разреженной и иерархической сети* (domain adaptation based on a sparse and hierarchical network, DASH-N). Наш метод комплексно изучает иерархию признаков вместе с преобразованиями, которые устраняют несоответствие между различными доменами. В основе DASH-N лежит скрытое разреженное представление. Мы используем этап уменьшения размерности, чтобы предотвратить слишком быстрое увеличение размерности данных по мере углубления в иерархию. Экспериментальные результаты показали, что этот метод выгодно отличается от конкурирующих современных методов разреженного обучения. Кроме того, показано, что многоуровневый DASH-N работает лучше, чем одноуровневый.

Скрытое разреженное представление: исходя из наблюдения, что сигналы часто присутствуют в низкоразмерном многообразии, в предыдущем разделе мы выполняли изучение словаря и разреженное кодирование в низкоразмерном скрытом пространстве. Чтобы облегчить дальнейшее обсуждение, мы определяем скрытое разреженное представление и соответствующую ему оптимизацию следующим образом:

$$L(Y, P, D, X) = \|PY - DX\|_F^2 + \alpha \|Y - P^T PY\|_F^2 + \beta \|X\|_1$$

$$\text{так, что } PP^T = I \text{ и } \|d_i\|_2 = 1, \forall i \in [1, K], \quad (7.11)$$

где $P \in R^{p \times d}$ – линейное преобразование, которое переводит данные в низкоразмерное пространство признаков ($p < d$). Заметим, что словарь теперь находится в низкоразмерном пространстве $D \in R^{p \times K}$. Первый член функции

потеря поощряет разреженность сигналов в пространстве уменьшенной размерности. Второй член – это количество энергии, отброшенной преобразованием P , или разница между низкоразмерными приближениями и исходными сигналами. Минимизация второго члена поощряет обученное преобразование сохранять полезную информацию, присутствующую в исходных сигналах. Помимо вычислительного преимущества, было показано (Nguyen et al., 2012), что эта оптимизация способна лучше восстанавливать базовое разреженное представление, чем традиционные методы обучения по словарю. Эта стратегия привлекательна, поскольку позволяет переносить данные в другой домен, чтобы лучше обрабатывать различные источники вариаций, такие как освещение и геометрические искажения.

Мы предлагаем метод для выполнения иерархической адаптации домена совместно с изучением признаков. На рис. 7.3 показана общая схема предлагаемого метода. Сеть содержит несколько уровней, каждый из которых включает в себя три подуровня, как показано на рис. 7.3. Первый подуровень выполняет нормализацию контраста и уменьшение размерности входных данных. На втором подуровне выполняется разреженное кодирование. В финальном подуровне к смежным признакам применяется тах-пулинг для создания нового признака. Выход одного слоя становится входом для следующего слоя. Для простоты обозначений мы рассматриваем один исходный домен. Расширение DASH-N на несколько исходных доменов можно выполнить с помощью процедуры, описанной в уравнении (7.8).

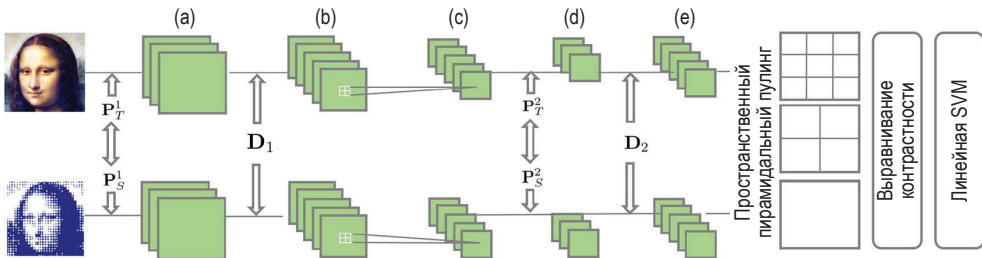


Рис. 7.3 ❖ Обзор алгоритма DASH-N (Nguyen et al., 2015). Сначала изображения делятся на небольшие перекрывающиеся участки. Эти участки векторизованы с сохранением их пространственного расположения. (а) Выполнение нормализации контраста и уменьшение размерности с использованием P_S для исходных изображений и P_T для целевых изображений. Обратные связи между P_S и P_T указывают на то, что эти два преобразования изучаются совместно. (b) Получение разреженных кодов с использованием общего словаря D_1 . (c) Выполнение тах-пулинга. Затем процесс повторяется для слоя 2 (этапы d и e), за исключением того, что входными данными являются разреженные коды из слоя 1, а не интенсивности пикселей. На заключительном этапе для создания дескрипторов изображений используется пространственная пирамида с тах-пулингом. Классификация выполняется с применением линейной машины опорных векторов на последнем слое

Пусть $Y_S \in R^{d \times N_S}$ и $Y_T \in R^{d \times N_T}$ будут входными данными на каждом уровне из исходного домена и целевого домена соответственно. Обратите внимание,

что у нас есть N_S , d -мерные образцы в исходном домене, и N_T , d -мерные образцы в целевом домене. Здесь мы предполагаем, что для простоты рассуждений исходные и целевые данные имеют одинаковую размерность d . Однако наша методика также подходит для сценария, в котором измерения данных различаются в разных доменах. При заданных Y_S и Y_T на каждом слое DASH-N мы изучаем совместное скрытое разреженное представление путем минимизации следующей функции потерь относительно (P_S, P_T, D, X_S, X_T) :

$$L(Y_S, P_S, D, X_S, \alpha, \beta) + \lambda L(Y_T, P_T, D, X_T, \alpha, \beta) \quad (7.12)$$

так, что $P_S P_S^T = P_T P_T^T = I$, $\|d_i\|_2 = 1$, $\forall i \in [1, K]$,

где (α, β, λ) – неотрицательные константы, $D \in R^{p \times K}$ – общий словарь, $P_S \in R^{p \times d}$ и $P_T \in R^{p \times d}$ – преобразования в скрытую область, $X_S \in R^{K \times N_S}$ и $X_T \in R^{K \times N_T}$ – разреженные коды источника и цели домена соответственно. Как видно из приведенной выше формулы, два домена вынуждены совместно использовать общий словарь в скрытом домене. Вместе с ограничением разреженности общий словарь D обеспечивает эффект связи, который способствует обнаружению общих структур для двух доменов. Для простоты далее мы подробно рассмотрим двухслойную сеть DASH-N. Расширение DASH-N на несколько слоев не составляет труда.

Слой 1: мы выполняем частую выборку на каждом обучающем изображении, чтобы получить набор перекрывающихся участков (патчей). Затем эти участки нормализуются по контрасту. Если f – вектор, соответствующий патчу, то нормализация контраста может быть записана следующим образом:

$$\hat{f} = \frac{f}{\sqrt{\|f\|^2 + \epsilon}}, \quad (7.13)$$

где ϵ – небольшая константа. Мы установили значение ϵ равным 0,1 просто потому, что в наших экспериментах оно хорошо работает. Чтобы сделать вычисления более эффективными, для изучения скрытого разреженного представления используется только случайное подмножество фрагментов каждого изображения. Мы обнаружили, что установка этого числа на 150 для изображений максимального размера 150×150 обеспечивает хороший компромисс между точностью и вычислительной эффективностью. После изучения словаря D_1 и преобразований (P_S^1, P_T^1) для всех выборочных патчей вычисляются разреженные коды (X_S^1, X_T^1) путем решения задачи

$$\min_{X_*^1} \|P_*^1 Y_*^1 - D_1 X_*^1\|_2^2 + \beta_1 \|X_*^1\|_1, \quad (7.14)$$

где символ $*$ указывает, что вышеуказанная задача (7.14) может соответствовать исходным или целевым данным. Каждый столбец Y_1 представляет собой векторизованные значения пикселей внутри патча. Для решения этой задачи оптимизации используется быстрая реализация алгоритма LARS (Mairal et al., 2009). Для объединения разреженных кодов по каждому соседству 4×4 используется пространственный max-пулинг, поскольку этот метод объеди-

нения особенно хорошо подходит для разделения разреженных признаков (Bureau, 2012; Bureau et al., 2010).

Слой 2: на этом уровне мы выполняем аналогичные вычисления, за исключением того, что входными данными являются разреженные коды из слоя 1, а не пиксели изображения. Признаки, полученные из предыдущего слоя, объединяются путем конкатенации по каждому соседству 4×4 и нормализуются по контрасту. Это дает нам новое представление, которое более устойчиво к окклюзии и изменениям освещения. Подобно слою 1, мы также случайным образом выбираем 150 нормализованных векторов признаков \hat{f} из каждого изображения для обучения. Далее ℓ_1 -оптимизация снова используется для вычисления разреженных кодов нормализованных признаков \hat{f} . В конце слоя 2 разреженные коды затем агрегируются с использованием тах-пулинга в многоуровневой декомпозиции патчей (т. е. пространственный пирамидальный тах-пулинг). На уровне 0 пространственной пирамиды один вектор признаков получается путем тах-пулинга всего изображения. На уровне 1 изображение делится на четыре квадранта, и к каждому квадранту применяется тах-пулинг, что дает 4 вектора признаков. Точно так же на уровне 2 мы получаем 9 векторов признаков и т. д. В этом примере мы рассматриваем тах-пулинг с использованием трехуровневой пространственной пирамиды. Таким образом, окончательный вектор признаков, возвращаемый вторым слоем для каждого изображения, является результатом объединения 14 векторов признаков из пространственной пирамиды.

Оптимизация. Рассмотрим более детально процедуру оптимизации целевой функции в уравнении (7.12). Во-первых, давайте определим

$$K_S = Y_S^T Y_S, \quad K_T = Y_T^T Y_T, \quad K = \begin{pmatrix} K_S & 0 \\ 0 & \sqrt{\lambda} K_T \end{pmatrix}, \quad (7.15)$$

$$K_S = V_S \Lambda_S V_S^T, \quad K_T = V_T \Lambda_T V_T^T$$

как матрицу Грама исходных данных, целевых данных и их блочно-диагональной конкатенации соответственно. Можно показать, что оптимальное решение (7.12) принимает следующий вид:

$$D = [A_S^T K_S, \sqrt{\lambda} A_T^T K_T] B, \quad P_S = (Y_S A_S)^T, \quad P_T = (Y_T A_T)^T \quad (7.16)$$

для некоторых $A_S \in R^{N_S \times p}$, $A_T \in R^{N_T \times p}$ и $B \in R^{(N_S + N_T) \times K}$. Мы можем подставить их в целевую функцию в уравнении (7.12) и оптимизировать ее относительно (A_S, A_T, B) . Обратите внимание, что строки каждого преобразования находятся в подпространстве столбцов данных из его собственного домена. Напротив, столбцы словаря создаются совместно данными как источника, так и цели. Когда (B, X) фиксированы, мы можем найти (A_S, A_T) , сначала решив следующую ограниченную оптимизацию:

$$\min_G \text{tr}(G^T H G) \text{ так, что } G_S^T G_S = G_T^T G_T = I, \quad G = [G_S, \sqrt{\lambda} G_T], \quad (7.17)$$

где

$$H = \Lambda^{0,5} V^T K ((I - BX)(I - BX)^T - \alpha I) K V \Lambda^{0,5}. \quad (7.18)$$

Тогда решения (A_S, A_T) имеют вид:

$$A_S = V_S \Lambda_S^{-0,5} G_S, \quad A_T = V_T \Lambda_T^{-0,5} G_T. \quad (7.19)$$

Когда (A_S, A_T) фиксированы, мы можем найти (B, X) , используя стандартную процедуру разреженного кодирования:

$$\|Z - DX\|_F^2 + \beta(\|X_S\|_1 + \lambda\|X_T\|_1) \quad (7.20)$$

так, что $Z = [A_S K_S, \sqrt{\lambda} A_T K_T]$, $X = [X_S, \sqrt{\lambda} X_T]$, $B = Z^+ D$.

Здесь Z^+ обозначает псевдоинверсию Мура–Пенроуза матрицы Z .

Эксперименты. Предлагаемый алгоритм оценивается в контексте распознавания объектов с использованием набора данных адаптации домена (Saenko et al., 2010), содержащего 31 класс, с добавлением изображений из набора данных Caltech-256 (Griffin et al., 2007). Смещения доменов вызваны различиями в таких факторах, как ракурс, освещение, разрешение и т. д. между изображениями в разных доменах. Кроме того, чтобы лучше оценить способность адаптироваться к широкому спектру доменов, мы также представляем экспериментальные результаты применения нашего подхода к новым изображениям, полученным путем выполнения алгоритмов полутонного изображения и обнаружения краев на изображениях из наборов данных в (Saenko et al., 2010 г.; Griffin et al., 2007).

Таблица 7.2. Показатели распознавания для различных подходов в четырех доменах (C: Caltech, A: Amazon, D: DSLR, W: Webcam). Используется 10 общих классов

Метод	C \uparrow A	C \uparrow D	A \uparrow C	A \uparrow W	W \uparrow C	W \uparrow A	D \uparrow A	D \uparrow W
Metric (Saenko et al., 2010)	33,7	35,0	27,3	36,0	21,7	32,3	30,3	55,6
SGF (Gopalan et al., 2011)	40,2	36,6	37,7	37,9	29,2	38,2	39,2	69,5
GFK (PLS+PCA) (Gong et al., 2012)	46,1	55,0	39,6	56,9	32,8	46,2	46,2	80,2
SDDL (Shekhar et al., 2013)	49,5	76,7	27,4	72,0	29,7	49,4	48,9	72,6
HMP (Manjunath, Chellappa, 1993)	67,7	70,2	51,7	70,0	46,8	61,5	64,7	76,0
DASH-N (1-й слой) (Nguyen et al., 2015)	60,3	79,6	52,2	74,1	45,3	68,7	65,9	76,3
DASH-N (Nguyen et al., 2015)	71,6	81,4	54,6	75,5	50,2	70,4	68,9	77,1

Результаты распознавания различных алгоритмов на 8 парах исходных/целевых доменов показаны в табл. 7.2. Видно, что DASH-N превосходит все сравниваемые методы в 7 из 8 пар исходных/целевых доменов. Для таких пар, как Caltech–Amazon, Webcam–Amazon или DSLR–Amazon, мы достигли преимущества более чем на 20 % по сравнению со следующим лучшим алгоритмом без использования в сравнении обучения признаков (с 49,5 до 71,6 %, с 49,4 % до 70,4 % и с 48,9 до 68,9 % соответственно). Стоит отметить,

что, хотя мы используем генеративный подход для изучения признака, наш метод неизменно обеспечивает лучшее качество, чем (Shekhar et al., 2013), который использует дискриминативное обучение вместе с нелинейными ядрами. Также из таблицы видно, что многослойный DASH-N превосходит однослойный DASH-N. В случае адаптации набора данных Caltech-Amazon прирост качества за счет комбинации признаков, полученных из обоих слоев, а не только признаков из первого слоя, составляет более 10 % (с 60,3 до 71,6 %).

7.3.3. Инкрементное изучение словаря для адаптации предметной области без учителя

Далее мы рассмотрим метод пошагового изучения словаря, в котором для облегчения адаптации выбираются некоторые целевые данные, называемые *вспомогательными образцами* (supportive samples), как показано на рис. 7.4. Инкрементный характер этого подхода позволяет пользователям выбирать различное количество вспомогательных образцов (в нашем случае – изображений) в соответствии со своим бюджетом или до тех пор, пока не будет достигнута удовлетворительная точность. Вспомогательные образцы близки к исходному домену и обладают двумя свойствами: во-первых, их предсказанные метки классов надежны и могут использоваться для построения моделей классификации с более четким разделением; во-вторых, они действуют как мост, соединяющий два домена и уменьшающий расхождение доменов. С теоретической точки зрения оба свойства важны для адаптации, поэтому имеет смысл вносить вспомогательные образцы в исходный домен. Чтобы добиться монотонного уменьшения несоответствия домена во время адаптации, разработан *критерий остановки*. Экспериментальные результаты применения данного метода на нескольких широко используемых наборах визуальных данных показывают, что он работает лучше, чем многие современные методы разреженного обучения.

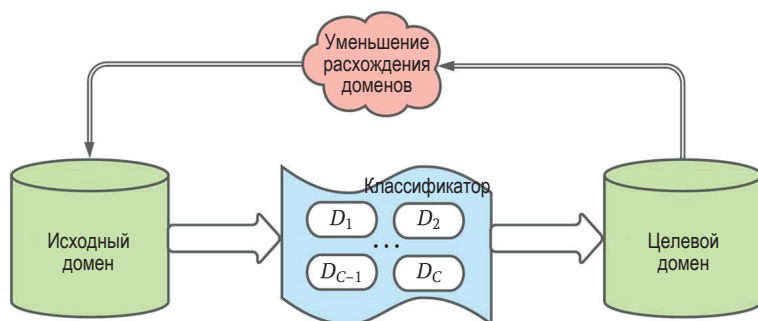


Рис. 7.4 ❖ Обобщенная схема инкрементного изучения словаря для адаптации домена. Исходные словари адаптируются к целевому домену с использованием набора вспомогательных образцов. Итерационный характер процедуры гарантирует монотонное уменьшение несоответствия доменов

Располагая словарем $D^{(k)}$, мы должны выбрать подмножество вспомогательных образцов. У нас есть два ограничения на этот выбор. Во-первых, следует исключить вспомогательные образцы, выбранные в предыдущих итерациях, поскольку мы хотим добавить новые данные для адаптации. Во-вторых, мы выбираем равное количество вспомогательных образцов для каждого класса, чтобы обеспечить баланс классов во время адаптации (Gong et al., 2013). Руководствуясь этими двумя ограничениями, мы выбираем наиболее надежные образцы, которые минимизируют ошибку реконструкции. Затем обновляем расширенный исходный домен, добавляя вспомогательные образцы, и переобучаем словарь. После этого мы проверяем критерий останова, чтобы увидеть, уменьшит ли добавление новых вспомогательных образцов расхождение доменов. Обобщенная схема предлагаемого подхода показана на рис. 7.4. Алгоритм состоит из следующих основных компонентов:

Обновление матрицы достоверности. Мы используем $X_S = X^{(0)}$ и X_T для обозначения данных из исходного и целевого доменов. Пусть $L = [1, \dots, C]$ представляет существующий набор меток. Пусть $D^{(0)} = [D_1^{(0)} \mid \dots \mid D_C^{(0)}]$ обозначает исходный словарь, обученный в исходном домене, где $D_j^{(0)}$ обозначает подсловарь, соответствующий классу j . Пусть $P \in R^{N \times C}$ обозначает матрицу достоверности, элементы которой $p_{ij} \in [0, 1]$ представляют вероятность того, что целевой образец x_i^t принадлежит классу j . В $(k+1)$ -й итерации мы обновляем матрицу достоверности $P^{(k+1)}$, используя текущие словари $D^{(k)} = [D_1^{(k)} \mid \dots \mid D_C^{(k)}]$ для конкретных классов.

$$p_{ij}^{(k+1)} = \begin{cases} \frac{2^{-0,5} \sigma \exp(-e_{ij}^{(k+1)}/2\sigma^2)}{\sum_{l=1}^C 2^{-0,5} \sigma \exp(-e_{il}^{(k+1)}/2\sigma^2)}, & \text{если } j = \operatorname{argmax}_l p_{il}^{(k+1)} \\ 0 & \text{в ином случае} \end{cases}, \quad (7.21)$$

где σ^2 – параметр нормализации, а e_{ij} обозначает ошибку реконструкции целевого образца x_i^t с использованием $D_j^{(k)}$:

$$e_{ij}^{(k+1)} = \|x_i^t - D_j^{(k)} \cdot Z_{ij}^{(k+1)}\|^2. \quad (7.22)$$

Здесь $Z_{ij}^{(k+1)}$ – разреженный код. Мы устанавливаем ограничение $p_{ij}^{(k+1)}$ только тогда, когда j является наиболее вероятным классом, к которому принадлежит образец i . Это ограничение гарантирует, что один и тот же образец не может быть выбран в качестве вспомогательного образца для нескольких классов.

Выбор вспомогательного образца. Мы выбираем новые вспомогательные образцы, используя $W^{(k+1)}$, для чего выполняем следующую оптимизацию:

$$W_j^{(k+1)} = \operatorname{argmax}_{W_j} \operatorname{tr}(W_j P_j^{(k+1)}) \quad (7.23)$$

$$\text{так, что } W_j^{(k+1)} \cdot \sum_{l=1}^k W_j^{(l)} = 0, \|W_j^{(k+1)}\|_0 = Q, j = 1, \dots, C, \quad (7.24)$$

где $W_j \in R^{N_i \times N_i}$ – диагональные матрицы, содержащие j -й столбец матрицы W на диагонали. То есть $W_j = \text{diag}([w_{1j}, w_{2j}, \dots])$. Аналогично, $P_j = \text{diag}([p_{1j}, p_{2j}, \dots])$. Q – количество вспомогательных образцов для каждого класса. Целевая функция уравнения (7.23) максимизирует достоверность выбранных вспомогательных образцов. Первое ограничение требует, чтобы вспомогательные образцы в $(k+1)$ -й итерации не пересекались с ранее выбранными, что гарантирует добавление новых вспомогательных образцов в исходный домен. Второе ограничение гарантирует, что количество вспомогательных образцов для каждого класса сбалансировано. Решение уравнения (7.23) заключается в нахождении соответствующих вспомогательных образцов Q , которые максимизируют достоверность с ограничением, заключающимся в том, что старые вспомогательные образцы исключаются.

Расширенное обновление исходного домена. После выбора вспомогательных образцов мы обновляем расширенные исходные данные, добавляя взвешенные вспомогательные образцы к предыдущим исходным данным:

$$X_j^{(k+1)} = [X_j^{(k)} | X^t W_j^{(k+1)} P_j^{k+1}], \quad k = 1, \dots, C. \quad (7.25)$$

Поскольку метки вспомогательных образцов могут содержать ошибки, каждый выбранный вспомогательный образец взвешивается по его достоверности. Веса указывают на надежность меток вспомогательных образцов, поэтому образцы с высокой степенью достоверности будут вносить больший вклад в модель.

Обновление словаря. Словарь обновляется путем решения следующей задачи оптимизации:

$$D_j^{(k+1)} = \underset{D_j, Z_j}{\operatorname{argmin}} \|X_j^{(k+1)} - D_j Z_j\|_F^2 + \lambda \|Z_j\|_1, \quad j = 1, \dots, C. \quad (7.26)$$

Мы решаем уравнение (7.26) с использованием метода изучения словаря из статьи (Mairal et al., 2009). Словарь, полученный на предыдущей итерации, используется в качестве начального словаря на следующей итерации. Таким образом, вычислительные затраты относительно невелики.

Критерий остановки. Один из тривиальных критериев остановки – отсутствие новых вспомогательных образцов для одного из классов. Но мы стремимся гарантировать, что процесс адаптации монотонно уменьшает расхождение доменов. Таким образом, ошибка классификации, связанная с целевым доменом, уменьшится, как указано в (Ben-David et al., 2010 г.; Smetana et al., 2009). Поэтому мы разработали меру сходства доменов, которую рассмотрим в следующем разделе, и выполняем адаптацию только тогда, когда сходство доменов увеличивается после каждой итерации. Можно показать, что при добавлении поддерживающих образцов в исходный домен сходство между исходным и целевым доменами будет увеличиваться. Читатели могут обратиться к (Lu et al., 2015) для более подробной информации о доказательстве этого свойства.

Эксперименты. Мы используем набор данных Office+Caltech, содержащий изображения из четырех доменов: Amazon (A), Webcam (W), DSLR (D) и Caltech (C). Это дает нам 12 пар доменов для тестирования. Во всех доменах

выбрано 10 общих классов. Для классов A, C, D и W имеется около 100, 100, 15 и 30 изображений соответственно. Мы следуем протоколу, предложенному в (Gong et al., 2013) для создания исходных и целевых данных домена. В нашем эксперименте используются признаки DeCAF (Donahue et al., 2014). Мы сравниваем два метода без адаптации (NA) и пять современных методов DA без учителя, а именно: SVM и классификация на основе словарного обучения (DLC) – это два метода NA, а интерполяция подпространства с помощью обучения по словарю (SIDL) (Ni et al. al., 2013), Geodesic Flow Kernel (GFK) (Gong et al., 2012), Transfer Joint Matching (TJM) (Long et al., 2014), Landmarks (Gong et al., 2012) и DA-NBNN (Tommasi, Caputo, 2013) относятся к методам DA с обучением без учителя. DLC реализован с использованием метода изучения словаря согласно (Mairal et al., 2009), а также применяется в качестве исходного словаря в предлагаемом нами методе.

Мы сравнили изменение сходства доменов на рис. 7.5 с результатами классификации в табл. 7.3 и обнаружили, что точность, судя по всему, увеличивается, когда значение сходства доменов продолжает расти по мере добавления дополнительных вспомогательных образцов к исходному домену.

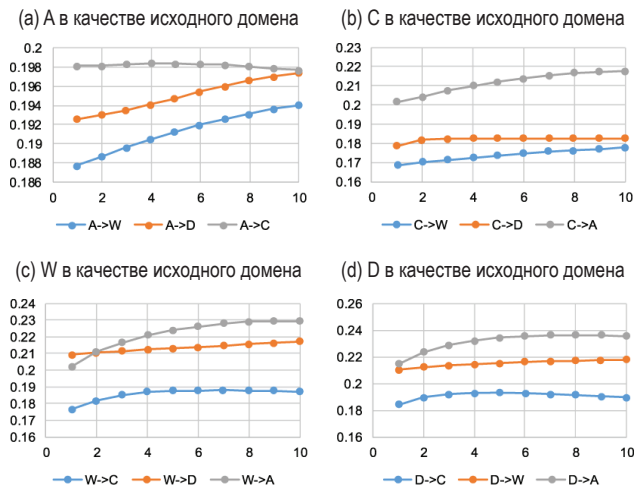


Рис. 7.5 ❖ Изменение сходства доменов при добавлении вспомогательных образцов к исходному домену. В наших экспериментах мы продолжаем адаптацию только до тех пор, пока значение сходства увеличивается. A: Amazon, C: Caltech, W: Webcam, D: DSLR

Мы показали, что изучение словаря является эффективным подходом к решению проблемы расхождения доменов с обучением без учителя. Общая идея состоит в том, чтобы проецировать представления данных из нескольких доменов в одно и то же скрытое пространство, где их распределения более похожи. Вместо того чтобы выполнять это преобразование на одном семантическом уровне, мы также продемонстрировали преимущества иерархического изучения словаря для постепенной адаптации на нескольких семантических уровнях. Наконец, мы демонстрируем преимущества по-

степенного добавления в исходный домен выборочных вспомогательных образцов, гарантированно увеличивающих сходство доменов, что обычно приводит к повышению точности классификации.

Таблица 7.3. Точность распознавания 12 междоменных пар объектов с обучением без учителя. A: Amazon, C: Caltech, W: Webcam, D: DSLR

	Метод	A \uparrow C	A \uparrow D	A \uparrow W	C \uparrow A	C \uparrow D	C \uparrow W	W \uparrow A	W \uparrow D	W \uparrow C	D \uparrow A	D \uparrow C	D \uparrow W
NA	SVM	85,0	87,9	79,0	91,4	89,8	80,0	75,7	99,4	72,0	87,1	78,8	98,6
	DLC	85,3	82,1	75,6	91,3	87,9	78,6	78,4	98,7	76,0	88,1	81,6	99,3
	GFK (Gong et al., 2012)	77,3	84,7	81,0	88,5	86,0	80,3	81,8	100	73,9	85,8	76,0	97,3
	SIDL (Ni et al., 2013)	84,5	81,5	74,2	90,9	89,8	78,3	75,1	100	71,1	87,9	80,1	99,3
	TJM (Long et al., 2014)	80,1	84,7	75,2	89,0	85,3	76,9	84,8	100	78,0	87,4	77,4	98,6
	DA-NBNN (Tommasi, Caputo, 2013)	83,4	80,9	76,6	89,6	87,9	80,3	88,0	100	82,4	91,3	86,1	98,0
DA	Landmarks (Gong et al., 2013)	84,7	86,0	82,4	92,4	92,3	84,1	84,0	98,7	71,7	77,0	74,4	95,2
	Online dictionary (Lu et al., 2015)	86,7	92,4	88,5	93,3	88,5	95,6	92,8	100	88,7	93,1	89,1	99,3

Подходы, основанные на словарях, похоже, потеряли популярность из-за более высокой точности методов, основанных на генеративно-состязательных сетях.

7.4. АДАПТАЦИЯ ДОМЕНА С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ СЕТЕЙ, ОБУЧАЕМЫХ БЕЗ УЧИТЕЛЯ

Глубокие нейронные сети – это мощный класс моделей машинного обучения для извлечения осмысленных представлений из изображений. Несмотря на достижение потрясающей точности в нескольких задачах визуального распознавания (Ren et al., 2015; He et al., 2016; 2017), нейронные сети очень чувствительны к расхождению доменов, т. е. точность нейронных сетей значительно снижается, когда тестовое распределение отличается от обучающего. Чтобы решить проблему смещения домена, в задаче обучения используются дополнительные функции потерь, предотвращающие дрейф исходного и целевого пространств признаков.

Схема адаптации домена с использованием глубоких нейросетей показана на рис. 7.6. Исходное и целевое распределения признаков сначала получают путем пропускания соответствующих изображений через сеть признаков F. Их распределения признаков выглядят непохожими из-за смещения

домена. Во время адаптации домена расстояние между этими пространствами признаков сводится к минимуму при одновременном обучении модели классификатора на размеченных исходных данных.

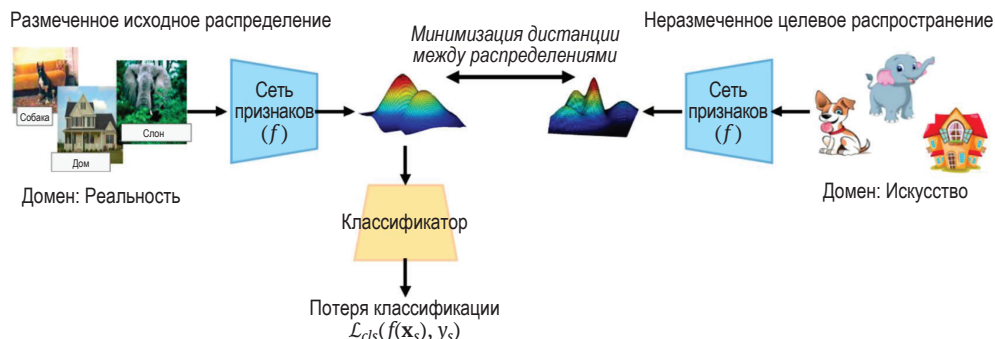


Рис. 7.6 ❖ Адаптация домена с использованием глубоких нейронных сетей. Исходное и целевое пространства объектов сближаются с использованием целевой функции минимизации расстояния между распределениями

Для решения задачи адаптации домена было предложено несколько дискриминационных и генеративных методов (Ganin et al., 2016; Long et al., 2016; 2017; Hoffman et al., 2018; Sankaranarayanan et al., 2018). В дискриминационных методах вместе с целевой функцией классификации обычно используют дополнительную функцию потерь, чтобы предотвратить дрейф пространства признаков. Эти функции обычно направлены на минимизацию расстояния между исходными и целевыми пространствами признаков (Ganin et al., 2016; Long et al., 2016; 2017). В генеративных подходах генеративная модель (обычно генеративно-сопоставительная сеть) обучается моделировать распределение исходных и целевых изображений. Эти знания об изученных распределениях затем используются для обеспечения инвариантности домена во время обучения (Hoffman et al., 2018; Sankaranarayanan et al., 2018).

7.4.1. Дискриминационные подходы к адаптации предметной области

Пусть F обозначает глубокую нейронную сеть для извлечения представлений признаков из изображений. Сеть F обычно является сверточной сетью, которая получает входные изображения и возвращает вектор на выходе. Пусть C обозначает сеть-классификатор, которая принимает представление признаков в качестве входных данных и возвращает логиты классификации. Пусть (x^s, y^s) обозначают пары вход–выход исходного домена, а (x^t) обозначают входы целевого домена.

Чтобы обучить модель классификатора в исходном домене, мы минимизируем кросс-энтропийную потерю как

$$L_{cls} = E[-y^s \log(C(F(x^s)))] \quad (7.27)$$

Результирующий классификатор будет плохо работать в целевом домене из-за расхождения доменов. Дрейф пространства признаков сводится к минимуму с помощью состязательных потерь домена, которые показывают, насколько различаются исходные и целевые представления признаков. Чтобы найти состязательную потерю домена, мы используем дополнительную сеть, называемую дискриминатором (D), как показано на рис. 7.7. Дискриминатор принимает в качестве входных данных представления признаков, полученные из сети F, и предсказывает, из какого домена они поступили – исходного или целевого. Затем выполняется состязательное обучение сети F, чтобы максимизировать потери дискриминатора. Считается, что обучение сошло, когда дискриминатор не справился со своей задачей, т. е. когда исходное и целевое распределения признаков неразличимы.

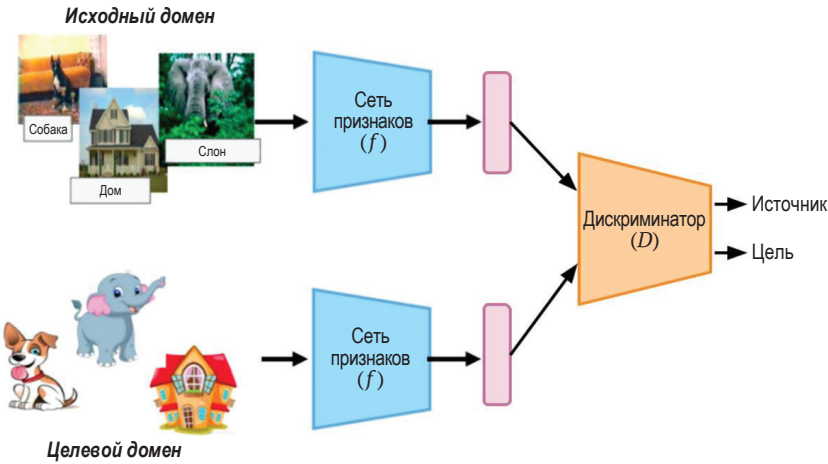


Рис. 7.7 ❖ Состязательное обучение при адаптации домена

Для достижения вышеуказанной цели сеть дискриминатора максимизирует следующую функцию потерь:

$$L_{disc} = \log[D(F(x^s))] + \log[1 - D(F(x^t))]. \quad (7.28)$$

Во время обучения модели оптимизируются с использованием комбинации потери классификации и состязательной потери домена:

$$\min_{F,C} \max_D [L_{cls} + \lambda L_{disc}]. \quad (7.29)$$

Член λ управляет весом, придаваемым состязательному члену домена в целевой функции. Это гиперпараметр, который необходимо настраивать при обучении алгоритма. Полученный алгоритм получил название Domain Adversarial Training (Ganin et al., 2016).

Сети F, C и D обычно реализуются как нейронные сети. В частности, сеть F – это глубокая сверточная сеть (такая как Resnet), а сети C и D – многослойные перцептроны. У Ганина и др. (Ganin et al., 2016) состязательная потеря реализована при помощи слоя обращения градиента, в котором градиенты, поступающие от сети дискриминатора с потерей L_{disc} , модулируются с коэффициентом $-\lambda$ перед обновлением сети признаков. Отметим, что сеть признаков также обновляется с использованием потери классификации. Все сети оптимизированы с использованием стохастического градиентного спуска.

Эффективность состязательной адаптации домена по сравнению с другими методами глубокой адаптации показана в табл. 7.4 для набора данных Office-31, который охватывает три домена – Amazon (A), DSLR (D) и Webcam (W). Задача состоит в том, чтобы выполнить классификацию 31 категории объектов. В табл. 7.4 мы видим, что состязательная адаптация домена обеспечивает значительный прирост точности по сравнению с базовыми моделями, которые обучаются только на исходном домене без какой-либо адаптации. Кроме того, модель также работает лучше, чем два других метода дискриминационной адаптации – Deep Domain Confusion (Tzeng et al., 2014) и Deep Adaptation Networks (Long et al., 2015).

Таблица 7.4. Точность состязательной адаптации домена (%) для набора данных Office-31. A: Amazon, D: DSLR, W: Webcam

Метод	A → W	D → W	W → D
Только источник	64,2	96,1	97,8
DDC (Tzeng et al., 2014)	61,8	95,0	98,5
DAN (Long et al., 2015)	68,5	96,0	99,0
Состязательное обучение (Ganin et al., 2016)	73,0	96,4	99,2

Другие меры расстояния. Несмотря на то что состязательное изучение домена является популярным выбором для дискриминационной адаптации, для измерения несоответствия распределения между исходным и целевым распределениями также можно использовать другие меры расстояния. Двумя такими мерами расстояния являются *максимальное среднее расхождение* (maximum mean discrepancy, MMD) и *расстояние Вассерштейна* (Wasserstein distance). В MMD расстояние между распределениями вычисляется как расстояние между средними представлениями, выраженными с использованием *воспроизводящего ядра гильбертова пространства* (reproducing kernel Hilbert space, RKHS). Затем выполняется адаптация путем минимизации MMD между исходным и целевым распределениями признаков (Long et al., 2015; 2016).

В альтернативном методе в качестве меры расстояния между исходным и целевым распределениями признаков используют расстояние Вассерштейна. Двойственная форма расстояния Вассерштейна вычисляется с использованием функции дискриминатора. Адаптация осуществляется путем минимизации двойственного расстояния Вассерштейна между исходными и целевыми картами признаков (Shen et al., 2018). Эти методы работают не хуже состязательной адаптации.

Метод псевдометок. В отличие от состязательных методов, при псевдомаркировке/самообучении (Saito et al., 2017; Zou et al., 2018; 2019) для неразмеченных целевых объектов генерируются псевдометки с использованием текущей модели. Поскольку псевдометки, как правило, зашумлены, наиболее надежные псевдометки выбирают с использованием показателей оценки достоверности. Одним из таких показателей является согласованность ансамбля классифицирующих моделей (Saito et al., 2017). Затем полученные псевдометки используются для переобучения модели с целью повышения ее прогностической способности в целевом домене. Этот процесс повторяется итеративно до сходимости. Данные методы также могут использоваться в сочетании с состязательной адаптацией.

Методы, основанные на регуляризации. Согласно таким методам, сети обучаются с дополнительным членом регуляризации наряду с потерей кросс-энтропии для исходного домена. Одной из распространенных функций регуляризации является минимизация энтропии (Vu et al., 2019), при которой минимизируется энтропия целевых логитов. Также заслуживают внимания регуляризация состязательного отсева (Saito et al., 2017) и максимальное расхождение классификатора (Saito et al., 2018), в котором сеть признаков обучается минимизировать расхождение в логитах, возникающее из различных моделей классификатора, которые обучаются в комбинации.

Расширение до адаптации с несколькими источниками. В адаптации с несколькими источниками цель состоит в том, чтобы адаптировать несколько исходных распределений к целевому распределению. Адаптация нескольких исходных доменов очень полезна на практике, поскольку наборы данных реального мира обычно содержат смесь нескольких скрытых доменов. Цель состязательного обучения может быть расширена до структуры с несколькими источниками с использованием k -стороннего классификатора домена, как это сделано в (Xu et al., 2018). В работе (Yang et al., 2020) при выборе лучших образцов исходного домена для адаптации к данному целевому домену применяется механизм взвешивания. В случаях, когда метки домена доступны, в многосторонней состязательной адаптации может использоваться исследование скрытого домена с целью извлечения неизвестных меток домена (Mancini et al., 2018).

7.4.2. Генеративные подходы к адаптации домена

В генеративных методах адаптации домена цель заключается в том, чтобы использовать генеративные модели для оценки исходного и целевого распределений. Затем обученные генеративные модели используются в процессе адаптации для изучения представлений, не зависящих от домена. Три популярных варианта генеративных моделей – это генеративно-состязательные сети (GAN) (Goodfellow et al., 2014), вариационные автоэнкодеры (VAE) (Kingma, Welling, 2013) и нормализующие потоки (Papamakarios et al., 2019). Сети GAN широко использовались для адаптации домена, поскольку они очень успешно генерировали образцы с высокой точностью.

Генеративные состязательные сети

Пусть $\{x_i\}_{i=1:N}$ будут образцами изображений, соответствующими входному распределению. Целью GAN (Goodfellow et al., 2014) является обучение модели, позволяющей генерировать образцы изображений, напоминающие входное распределение. Для этого GAN обучает модель G , которая отображает выборки из скрытого пространства в пространство входных изображений. Скрытое пространство обычно моделируется с использованием известного управляемого распределения, такого как многомерное изотропное гауссово распределение. Как только модель G обучена, мы можем синтезировать изображения, делая выборки из скрытого распределения и подавая их на вход генеративной модели G .

Для обучения генератора G мы используем вторую модель, называемую дискриминатором (D), или критикующей сетью. Назначение дискриминатора состоит в том, чтобы различать изображения, поступающие из реального и сгенерированного распределений. Это задача бинарной классификации, когда реальные образцы рассматриваются как один класс, а сгенерированные образцы рассматриваются как второй класс. Модель генератора G обучается так, чтобы дискриминатор не справился с задачей различения. Модели G и D реализованы с использованием глубоких нейронных сетей, как показано на рис. 7.8. Следовательно, целевая функция GAN может быть записана следующим образом:

$$\min_G \max_D [E_{x \sim p_{data}} \log D(x) + E_{z \sim p_z} \log (1 - D(G(z)))]. \quad (7.30)$$

Член внутри квадратных скобок является отрицательным значением потери бинарной классификации для классификации образцов, взятых из реального распределения p_{data} или сгенерированного распределения $G(p_z)$. В то время как дискриминатор D стремится максимизировать целевую функцию, генератор минимизирует ее, что приводит к минимаксной игре для двух игроков. Общая цель состоит в том, чтобы найти седловую точку в этой игре. При сходимости генератор синтезирует реалистичные изображения, а дискриминатор не может различить реальное и сгенерированное распределения.

Сети GAN очень сложно обучать из-за минимаксного характера целевой функции. На практике используется несколько приемов стабилизации, чтобы заставить GAN сходиться к хорошим решениям (Liu et al., 2020). Например, к этим приемам относятся использование расстояния Вассерштейна (Arjovsky et al., 2017), применение спектральной нормализации в архитектуре сети (Miyato et al., 2018), методы регуляризации, такие как уменьшение веса (Liu et al., 2020), штраф за градиент (Gulrajani et al., 2017) и потери при сопоставлении признаков (Liu et al., 2020).

Условный синтез изображений. В предыдущем разделе мы сосредоточились на безусловном синтезе изображений, целью которого было просто генерировать изображения, напоминающие образцы изображений на входе. В условном синтезе изображений мы заинтересованы в создании выборок,

обусловленных некоторыми интересующими нас переменными. Одним из примеров является синтез, обусловленный классом, когда требуется генерировать изображения, принадлежащие одному конкретному классу. Когда обуславливающая переменная сама является изображением, это задача *трансляции изображения в изображение*. Но в данном случае нас интересует трансляция изображения, принадлежащего одному домену, в изображение, принадлежащее другому домену. В условном синтезе изображения обуславливающая переменная используется в качестве входных данных в дополнение к скрытому вектору.

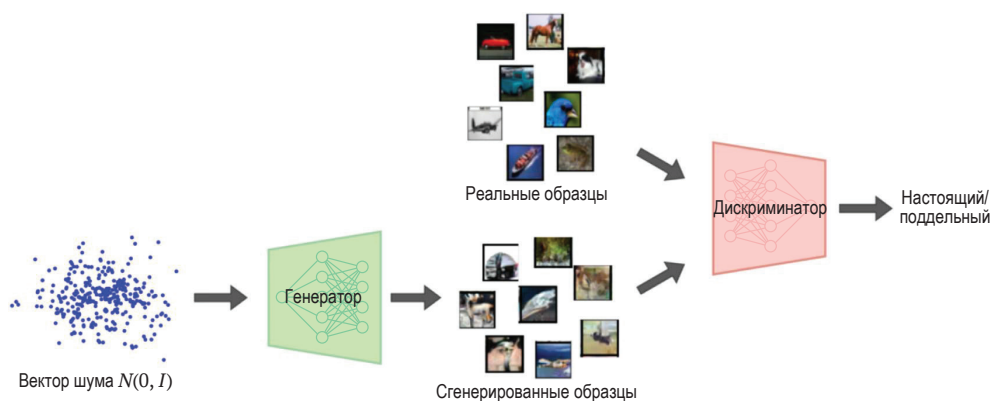


Рис. 7.8 ❖ Структура GAN

Когда речь идет об адаптации домена, модели преобразования изображения в изображение являются одним из предпочтительных вариантов генеративных моделей. Поскольку нас интересует адаптация домена на основе обучения без учителя, мы используем непарные модели перевода изображения в изображение, в которых исходное и целевое изображения не имеют никаких взаимных соответствий. Идея состоит в том, чтобы транслировать изображения исходного домена так, чтобы они выглядели как изображения целевого домена, а затем обучить классификатор на транслированных образцах исходного домена, используя метки источника в качестве эталона. Затем обученный классификатор можно использовать для тестового прогнозирования в целевом домене.

CycleGAN: это популярная модель для непарного преобразования изображения в изображение. Пусть X и Y обозначают два домена, между которыми должна транслировать изображения обучаемая модель. В CycleGAN мы используем две модели – прямую модель G , которая переводит изображения из домена X в домен Y , и обратную модель F , которая переводит изображения из домена Y в домен X . Дискриминаторная сеть D_X используется для получения состязательных потерь, чтобы гарантировать, что выборки, созданные сетью F , неотличимы от реального распределения X . Точно так же второй дискриминатор D_Y способствует тому, чтобы выборки, созданные G , были неотличимы от домена Y . На рис. 7.9 показано строение CycleGAN.

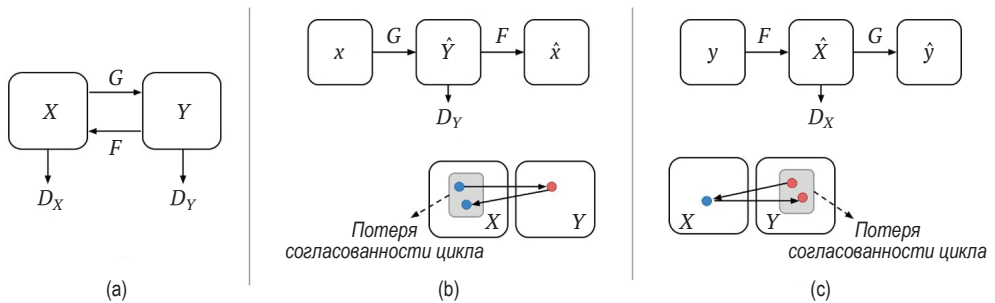


Рис. 7.9 ❖ (a) Структура CycleGAN, (b) состязательная потеря и потеря согласованности цикла для прямой модели, (c) состязательная потеря и потеря согласованности цикла для обратной модели

В то время как потери дискриминатора способствуют реалистичности сгенерированных выборок, в целевую функцию не входит член принудительного сохранения контента. То есть невозможно предотвратить сопоставление выборки из одного домена (например, все изображения кошек) с другим семантическим классом в другом домене (например, изображения собак). Чтобы предотвратить это, в задаче используется член согласованности цикла. Идея состоит в том, чтобы гарантировать, что применение прямой и обратной карт признаков к одному и тому же образцу возвращает входные данные, т. е. $F(G(x)) \approx x$ и $G(F(y)) \approx y$. Для согласованности цикла обычно используют потери L1.

$$L_{\text{cyc}}(G, F) = E_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|] + E_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|]. \quad (7.31)$$

Модель CycleGAN обучается с использованием комбинации состязательных потерь и потерь согласованности цикла.

$$L = L_{\text{adv1}} + L_{\text{adv2}} + \lambda L_{\text{cyc}}. \quad (7.32)$$

В ходе эксперимента мы обучали CycleGAN на наборах данных из следующих доменов: лошади и зебры, зима и лето, аэрофотоснимки и карты Google. Мы установили, что изображения переводятся из одного домена в другой с сохранением содержимого. На рис. 7.10 показана трансляция изображения из аэрофотоснимка в карту Google. Реконструкция $F(G(x))$ хорошо аппроксимирует входное изображение x , что подтверждает значимость члена согласованности цикла для сохранения содержимого. Кроме того, сгенерированные выборки реалистичны, что показывает эффективность состязательных потерь.

Адаптация домена на основе CycleGAN. В предыдущем разделе мы обсудили модель CycleGAN и то, как ее можно использовать для непарного преобразования изображения в изображение. Теперь мы покажем, как эту модель можно использовать для адаптации домена с обучением без учителя (Hoffman et al., 2018).

Сначала модель CycleGAN обучается трансляции изображений между исходным и целевым доменами. Затем модель задачи (модель классификации/

сегментации) обучается на транслированных исходных изображениях с использованием исходных меток в качестве исходных данных. Чтобы модель задачи хорошо обучалась, на картах признаков транслированных исходных изображений и истинных целевых изображений используется дополнительная потеря дискриминатора. Это гарантирует, что распределение признаков транслированных исходных изображений и целевых изображений будет совпадать. Схема этого фреймворка, также известного как CyCADA (Hoffman et al., 2018), изображена на рис. 7.11.

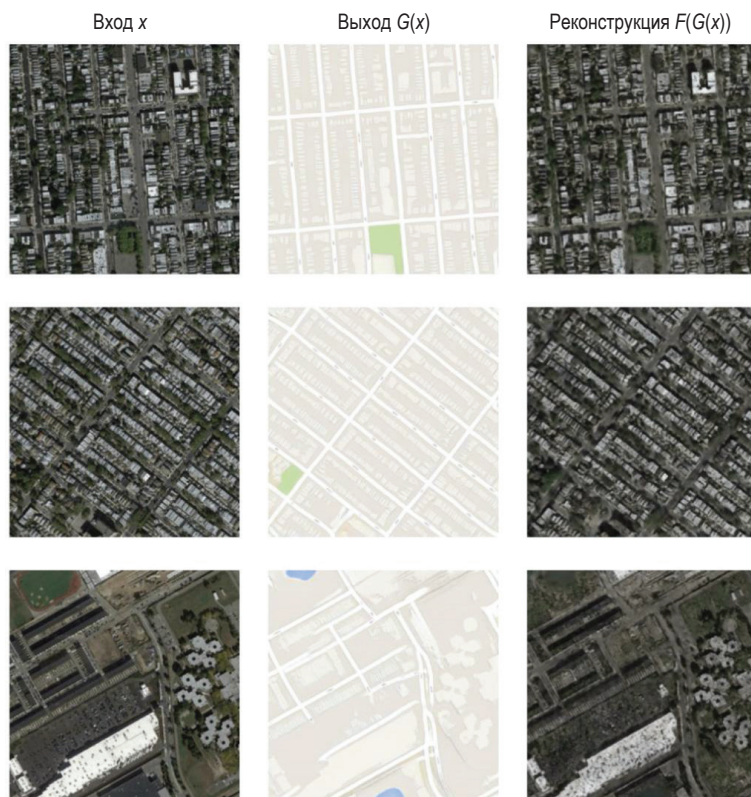


Рис. 7.10 ❖ Результаты работы CycleGAN. Входные изображения показаны на левой панели, а прямой и обратный переносы между доменами показаны на средней и правой панелях соответственно

Минимизация дистрибутивного расстояния с помощью генеративных моделей. Альтернативный подход к генеративной адаптации домена заключается в использовании GAN для минимизации дистрибутивного расстояния. В разделе 7.4.1 мы говорили, что адаптация домена формулируется как задача минимизации дистрибутивного расстояния между исходным и целевым распределениями признаков. Вместо того чтобы напрямую выполнять минимизацию расстояния в пространстве признаков, было предложено спроецировать признак обратно в пространство пикселей с помощью

GAN и выполнить минимизацию дистрибутивного расстояния в этом пространстве проецируемого изображения (рис. 7.12). Авторы метода (Sankaranarayanan et al., 2018) назвали этот подход Generate to Adapt.

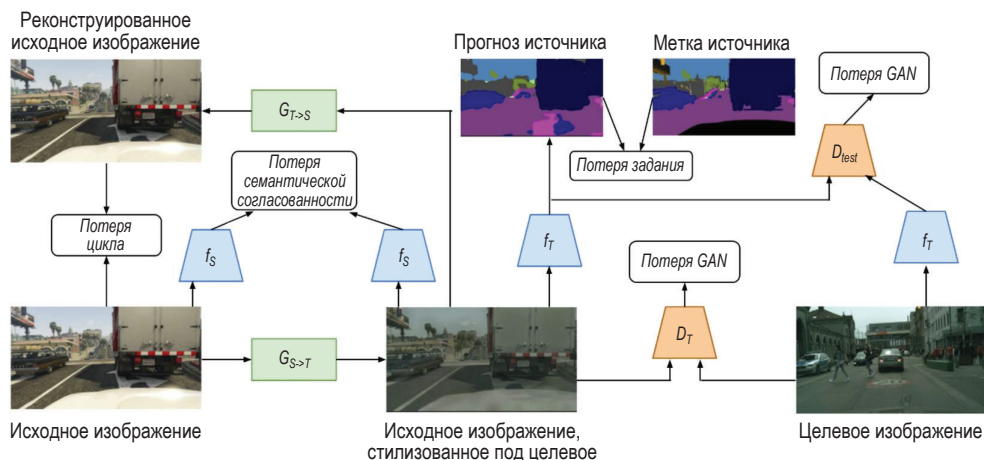


Рис. 7.11 ❖ Схема фреймворка SyCADA

Проецирование признаков обратно в пространство изображения имеет два основных преимущества: во-первых, пропускная способность сети дискриминатора эффективно увеличивается благодаря этапу проецирования. Во-вторых, этап проецирования помогает сохранить семантическое содержание в сгенерированных признаках. Иначе говоря, это предотвращает сопоставление целевых признаков из одного класса с другим классом, поскольку реконструирующая сеть обеспечивает обучающий сигнал при сохранении семантического содержания.

Структура фреймворка Generate to Adapt показана на рис. 7.12. Признаки исходного и целевого изображений извлекаются с помощью сети признаков F . Полученные признаки затем передаются через два потока: первый поток – это ветвь классификации, которая обучается с использованием перекрестной энтропийной потери в исходном домене. Второй поток – это ветвь минимизации расстояния. В этом потоке исходные и целевые признаки сначала инвертируются обратно в пространство изображения с использованием сети генератора G . Затем сгенерированные изображения проходят через дискриминатор, который различает, получены ли эти изображения из реального (исходного) или поддельного (целевого) домена. Кроме того, он также выполняет прогнозирование меток классов для реконструированных исходных изображений, используя исходные метки в качестве эталона. Это помогает достичь согласованности классов в реконструированных изображениях.

Затем сеть признаков F подвергается состязательному обучению, чтобы целевые реконструкции выглядели как исходные. Это происходит только тогда, когда исходное и целевое распределения признаков перекрываются. Кроме того, сигнал, полученный из потока 1, помогает сети признаков получать согласованные по классам прогнозы.

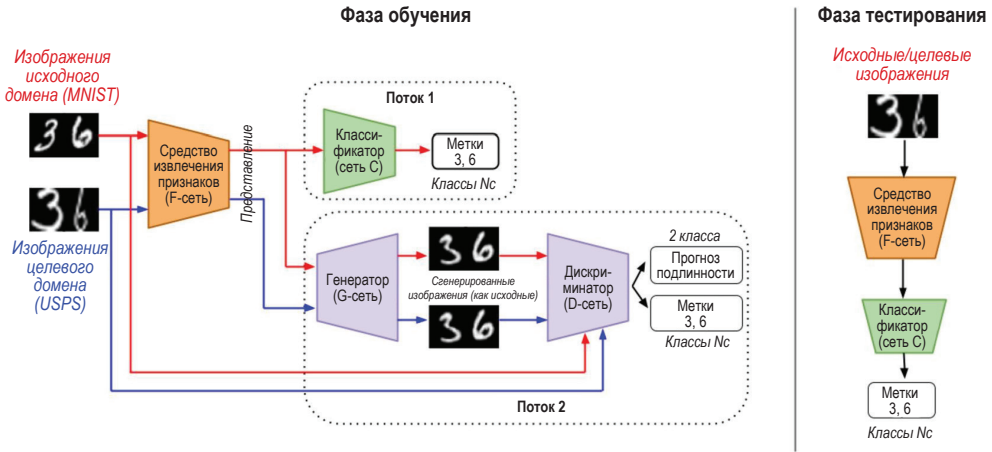


Рис. 7.12 ❖ Схема фреймворка Generate to Adapt

Результаты. В табл. 7.5 представлены результаты генеративных подходов к задачам междоменной классификации с использованием наборов изображений цифр. В экспериментах использовались три набора данных: MNIST, USPS и SVHN. В каждом из вариантов адаптации мы наблюдаем, что исходный базовый уровень обеспечивает низкую точность. И CyCADA, и Generate to Adapt обеспечивают значительный прирост точности по сравнению с базовой моделью. Кроме того, они также превосходят CoGAN и PixelDA, два других подхода к адаптации на основе GAN.

Таблица 7.5. Качество адаптации домена с использованием генеративных подходов к набору данных Digits. Критерием является точность классификации в %.

MN – MNIST, US – USPS, SV – SVHN

Метод	MS → US	US → MN	SV → MN
Базовая модель	79,1	57,1	60,3
CoGAN (Liu and Tuzel, 2016)	61,8	95,0	98,5
PixelDA (Bousmalis et al., 2017)	73,0	96,4	99,2
CyCADA (Hoffman et al., 2018)	95,6	96,5	90,4
Generate to Adapt (Sankaranarayanan et al., 2018)	92,8	95,3	92,4

В табл. 7.6 представлены результаты выполнения задачи междоменной семантической сегментации. Исходным доменом является GTA-5, который представляет собой синтетический набор данных уличных сцен, а целевым доменом является реальный набор данных Cityscapes (городские пейзажи). Мы использовали архитектуру FCN на основе VGG для сети признаков (Sankaranarayanan et al., 2018). Приведены показатели пересечения объединений (intersection of union, IoU) для каждого семантического класса вместе со средними показателями IoU (mIoU). Мы наблюдали, что как CyCADA, так и Generate to Adapt достигают значительного увеличения IoU по сравнению с исходной базовой моделью. Однако остается огромный разрыв по сравнению

Таблица 7.6. Производительность семантической сегментации в адаптации GTA-5 > Cityscapes. Сообщаются баллы IoU для каждой категории и средние баллы IoU (mIoU)

Метод	Дорога	Группа	Здание	Стена	Забор	Pole	Фонарь	Знак	Растения	Почва	Небо	Человек	Наездник	Автомобиль	Грузовик	Автобус	Трамвай	Мотоцикл	Велосипед	mIoU
Только источник	73,5	21,3	72,3	18,9	14,3	12,5	15,1	5,3	77,2	17,4	64,3	43,7	12,8	75,4	24,8	7,8	0,0	4,9	1,8	29,6
CyCADA	79,1	33,1	77,9	23,4	17,3	32,1	33,3	31,8	81,5	26,7	69,0	62,8	14,7	74,5	20,9	25,6	6,9	18,8	20,4	39,5
Generate to Adapt	88,0	30,5	78,6	25,2	23,5	16,7	23,5	11,6	78,7	27,2	71,9	51,3	19,5	80,4	19,8	18,3	0,9	20,8	18,4	37,1
Только цель	96,5	74,6	86,1	37,1	33,2	30,2	39,7	51,6	87,3	52,6	90,4	60,1	31,7	88,4	54,9	52,3	34,7	33,6	59,1	57,6

с целевой моделью, которая представляет собой модель оракула, обученную с использованием истинных целевых меток.

В табл. 7.7 представлено сравнение подходов на основе многообразий, словаря и GAN для адаптации домена без учителя в наборе данных Office. Методы на основе GAN обеспечивают наилучшее качество.

Таблица 7.7. Сравнение методов многообразий, словаря и глубокого обучения для адаптации домена без учителя

Источник	Цель	Многообразия (2012)	Словари (2015)	Глубокое обучение (2017)	Глубокое обучение и GAN (2018)
Webcam	DSLR	71,2		99,5	99,8
DSLR	Webcam	68,8	72	98,2	97,9
Amazon	Webcam	55,6	72	62,4	86,5
Amazon	DSLR			64	87,7
DSLR	Amazon		48,9	52	72,8
Webcam	Amazon		49,4	48,4	71,4

7.5. ЗАКЛЮЧЕНИЕ

В этой главе мы обсудили методы решения задачи адаптации домена, основанные на дифференциальной геометрии, разреженном представлении и глубоких нейронных сетях. Были рассмотрены два широких класса методов: дискриминативные и генеративные. В дискриминативных методах мы обучаем модель классификатора, используя дополнительные потери, чтобы сделать исходные и целевые распределения признаков похожими. Для этой задачи используется целевая функция минимизации дистрибутивного расстояния. В генеративных методах мы применяем для адаптации домена генеративную модель. Один из подходов заключается в обучении промежуточных словарей и междоменной GAN для сопоставления изображений из исходного и целевого доменов и обучения модели классификатора на преобразованных целевых изображениях. Другой подход основан на минимизации дистрибутивного расстояния и применяет GAN для минимизации этого расстояния. Все эти подходы проверены на задачах междоменного распознавания и семантической сегментации.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Aharon M., Elad M., Bruckstein A., 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* 54 (11), 4311–4322.
- Arjovsky M., Chintala S., Bottou L., 2017. Wasserstein generative adversarial networks. In: *International Conference on Machine Learning*, pp. 214–223.

- Ben-David S., Blitzer J., Crammer K., Kulesza A., Pereira F., Vaughan J. W., 2010. A theory of learning from different domains. *Machine Learning* 79 (1), 151–175.
- Bo L., Xiaofeng R., Dieter F., 2011. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In: *Advances in Neural Information Processing Systems*, vol. 24.
- Boureau Y. L., Ponce J., LeCun Y., 2010. A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 111–118.
- Bousmalis K., Silberman N., Dohan D., Erhan D., Krishnan D., 2017. Unsupervised pixel-level domain adaptation with generative adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3722–3731.
- Blitzer J., Crammer K., Kulesza A., Pereira F., Wortman J., 2008. Learning bounds for domain adaptation.
- Blitzer J., Kakade S., Foster D., 2011. Domain adaptation with coupled subspaces. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 173–181. *JMLR Workshop and Conference Proceedings*.
- Boureau Y. L., 2012. Learning hierarchical feature extractors for image recognition. Doctoral dissertation. New York University.
- Bruckstein A. M., Donoho D. L., Elad M., 2009. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review* 51 (1), 34–81.
- Chen S. S., Donoho D. L., Saunders M. A., 2001. Atomic decomposition by basis pursuit. *SIAM Review* 43 (1), 129–159.
- Daume III H., Marcu D., 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26, 101–126.
- Donahue J., Jia Y., Vinyals O., Hoffman J., Zhang N., Tzeng E., Decaf Darrell T., 2014. A deep convolutional activation feature for generic visual recognition. In: *International Conference on Machine Learning*, pp. 647–655. *PMLR*.
- Duan L., Tsang I. W., Xu D., Chua T. S., 2009. Domain adaptation from multiple sources via auxiliary classifiers. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 289–296.
- Duan L., Xu D., *Exploiting Web Chang S. F.*, 2012. Images for event recognition in consumer videos: a multiple source domain adaptation approach. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, pp. 1338–1345.
- Elad M., Figueiredo M. A., Ma Y., 2010. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE* 98 (6), 972–982.
- Engan K., Aase S. O., Husoy J. H., 1999. Method of optimal directions for frame design. In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing*. In: *Proceedings ICASSP99*, vol. 5. IEEE, pp. 2443–2446. (Cat. No. 99CH36258.)
- Ganin Y., Ustinova E., Ajakan H., Germain P., Larochelle H., Laviolette F., Marchand M., Lempitsky V., 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17 (1), 2096–2130.
- Gong B., Grauman K., Sha F., 2013. Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation. In: *International Conference on Machine Learning*, pp. 222–230. *PMLR*.

- Gong B., Shi Y., Sha F., Grauman K., 2012. Geodesic Flow Kernel for Unsupervised Domain Adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 16. IEEE, pp. 2066–2073.
- Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014. Generative adversarial networks. *arXiv preprint. arXiv: 1406.2661*.
- Gopalan R., Li R., Chellappa R., 2011. Domain adaptation for object recognition: an unsupervised approach. In: *2011 International Conference on Computer Vision*. IEEE, pp. 999–1006.
- Gopalan R., Li R., Chellappa R., 2014. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 2288–2302.
- Griffin G., Holub A., Perona P., 2007. Caltech-256 object category dataset.
- Gross R., Matthews I., Cohn J., Kanade T., Baker S., 2010. Multi-pie. *Image and Vision Computing* 28 (5), 807–813.
- Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A., 2017. Improved Training of Wasserstein gans. *arXiv preprint. arXiv:1704.00028*.
- He K., Zhang X., Ren S., Sun J., 2016. DeepResidual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- He K., Gkioxari G., Dollár P., Girshick R., 2017. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969.
- Ho H. T., Gopalan R., 2014. Model-driven domain adaptation on product manifolds for unconstrained face recognition. *International Journal of Computer Vision* 109, 110–125.
- Hoffman J., Tzeng E., Park T., Zhu J. Y., Isola P., Saenko K., Efros A., Darrell T., 2018. Cycada: cycle-consistent adversarial domain adaptation. In: *International Conference on Machine Learning*, pp. 1989–1998.
- Jhuo I. H., Liu D., Lee D. T., Chang S. F., 2012. Robust Visual domain adaptation with low-rank reconstruction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, pp. 2168–2175.
- Kingma D. P., Welling M., 2013. Auto-encoding variational Bayes. *arXiv preprint. arXiv:1312.6114*.
- Kulis Brian, Saenko Kate, Darrell Trevor, 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: *CVPR 2011*. IEEE, pp. 1785–1792.
- Lee J., Moghaddam B., Pfister H., Machiraju R., 2005. A bilinear illumination model for robust face recognition. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, vol. 2. IEEE, pp. 1177–1184.
- Liu M. Y., Tuzel O., 2016. Coupled generative adversarial networks. *arXiv preprint. arXiv:1606.07536*.
- Liu M. Y., Huang X., Yu J., Wang T. C., Mallya A., 2020. Generative adversarial networks for image and video synthesis: algorithms and applications. *arXiv preprint. arXiv:2008.02793*.
- Long M., Wang J., Ding G., Sun J., Yu P. S., 2014. Transfer joint matching for unsupervised domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1410–1417.

- Long M., Cao Y., Wang J., Jordan M., 2015. Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning, pp. 97–105.
- Long M., Zhu H., Wang J., Jordan M. I., 2016. Unsupervised domain adaptation with residual transfer networks. arXiv preprint. arXiv:1602.04433.
- Long M., Zhu H., Wang J., Jordan M. I., 2017. Deep transfer learning with joint adaptation networks. In: International Conference on Machine Learning, pp. 2208–2217.
- Lu B., Chellappa R., Nasrabadi N. M., 2015. Incremental dictionary learning for unsupervised domain adaptation. In: BMVC, pp. 108.1–108.12.
- Lui Y. M., Beveridge J. R., Kirby M., 2010. Action classification on product manifolds. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 833–839.
- Lui Y. M., 2012. Human gesture recognition on product manifolds. Journal of Machine Learning Research 13 (1), 3297–3321.
- Mairal J., Bach F., Ponce J., Sapiro G., 2009. Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 689–696.
- Mairal J., Bach F., Ponce J., 2011. Task-driven dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (4), 791–804.
- Mancini M., Porzi L., Bulò S. R., Caputo B., Ricci E., 2018. Boosting domain adaptation by discovering latent domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3771–3780.
- Manjunath B. S., Chellappa R., 1993. A unified approach to boundary perception: edges, textures, and illusory contours. IEEE Transactions on Neural Networks 4 (1), 96–108.
- Mansour Y., Mohri M., Rostamizadeh A., 2009. Domain adaptation: learning bounds and algorithms. arXiv preprint. arXiv:0902.3430.
- Miyato T., Kataoka T., Koyama M., Yoshida Y., 2018. Spectral normalization for generative adversarial networks. arXiv preprint. arXiv:1802.05957.
- Nguyen H. V., Ho H. T., Patel V. M., Chellappa R., 2015. DASH-n: joint hierarchical domain adaptation and feature learning. IEEE Transactions on Image Processing 24 (12), 5479–5491.
- Nguyen H. V., Patel V. M., Nasrabadi N. M., Chellappa R., 2012. Sparse embedding: a framework for sparsity promoting dimensionality reduction. In: Proceedings of the European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 414–427.
- Ni J., Qiu Q., Chellappa R., 2013. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, pp. 692–699.
- Olshausen B. A., Field D. J., 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381 (6583), 607–609.
- Pan WeiKe, Evan Xiang, Nathan Liu, Qiang Yang, 2010. Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 24(1).
- Papamakarios G., Nalisnick E., Rezende D. J., Mohamed S., Lakshminarayanan B., 2019. Normalizing flows for probabilistic modeling and inference. arXiv preprint. arXiv:1912.02762.

- Park Sung Won, Savvides Marios*, 2011a. The multifactor extension of Grassmann manifolds for face recognition. In: 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG). IEEE, pp. 464–469.
- Park Sung Won, Savvides Marios*, 2011b. Multifactor analysis based on factor-dependent geometry. In: CVPR 2011. IEEE, pp. 2817–2824.
- Patel V. M., Gopalan R., Li R., Chellappa R.*, 2015. Visual domain adaptation: a survey of recent advances. IEEE Signal Processing Magazine 32 (3), 53–69.
- Patil Y. C., Rezaiifar R., Krishnaprasad P. S.*, 1993. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers. IEEE, pp. 40–44.
- Ren S., He K., Girshick R., Sun J.*, 2015. Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint. arXiv:1506.01497.
- Roweis S. T., Saul L. K.*, 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500), 2323–2326.
- Rubinstein R., Bruckstein A. M., Elad M.*, 2010. Dictionaries for sparse representation modeling. Proceedings of the IEEE 98 (6), 1045–1057.
- Saenko K., Kulis B., Fritz M., Darrell T.*, 2010. Adapting visual category models to new domains. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 213–226.
- Saito K., Ushiku Y., Harada T.*, 2017a. Asymmetric tri-training for unsupervised domain adaptation. In: International Conference on Machine Learning, pp. 2988–2997.
- Saito K., Ushiku Y., Harada T., Saenko K.*, 2017b. Adversarial dropout regularization. arXiv preprint. arXiv:1711.01575.
- Saito K., Watanabe K., Ushiku Y., Harada T.*, 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732.
- Sankaranarayanan S., Balaji Y., Chellappa R.*, 2018. Adapting across Domains Using Generative Adversarial Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Spotlight paper). Salt Lake City, UT.
- Sankaranarayanan S., Balaji Y., Chellappa R.*, 2008. Learning from Synthetic Data: Semantic Segmentation across Domain Shift. (Spotlight Paper). In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT.
- Sharma A., Jacobs D. W.*, 2011. Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch. In: CVPR 2011. IEEE, pp. 593–600.
- Sharma A., Kumar A., Daume H., Jacobs D. W.*, 2012. Generalized multiview analysis: a discriminative latent space. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2160–2167.
- Shekhar S., Patel V. M., Nguyen H. V., Chellappa R.*, 2013. Generalized domain-adaptive dictionaries. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 361–368.
- Shen J., Qu Y., Zhang W., Wasserstein Yu Y.*, 2018. Distance guided representation learning for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (1).

- Shi Y., Sha F., 2012. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: Proceedings of International Conference on Machine Learning, pp. 1079–1086.
- Smetana Judith G., Villalobos Myriam, Tasopoulos-Chan Marina, Gettman Denise C., Campione-Barr Nicole, 2009. Early and middle adolescents' disclosure to parents about activities in different domains. *Journal of Adolescence* 32 (3), 693–713.
- Tenenbaum J. B., De Silva V., Langford J. C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290 (5500), 2319–2323.
- Tommasi T., Caputo B., 2013. Frustratingly easy domain adaptation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 897–904.
- Tropp J. A., 2004. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory* 50 (10), 2231–2242.
- Tzeng E., Hoffman J., Zhang N., Saenko K., Darrell T., 2014. Deep domain confusion: maximizing for domain invariance. *arXiv preprint. arXiv:1412.3474*.
- Vageeswaran P., Mitra K., Chellappa R., 2013. Blur and illumination robust face recognition via set-theoretic characterization. *IEEE Transactions on Image Processing* 22 (4), 1362–1372.
- Vu T. H., Jain H., Bucher M., Cord M., Pérez P., 2019. Advent: adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2517–2526.
- Vasilescu M. A., Terzopoulos D., 2002. Multilinear analysis of image ensembles: tensorfaces. In: *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, pp. 447–460.
- Vasilescu M. A., Terzopoulos D., 2007. Multilinear projection for appearance-based recognition in the tensor framework. In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE, pp. 1–8.
- Wang C., Mahadevan S., 2009. Manifold alignment without correspondence. In: *Twenty-First International Joint Conference on Artificial Intelligence*, p. 26.
- Wen Z., Yin W., 2013. A feasible method for optimization with orthogonality constraints. *Mathematical Programming* 142 (1), 397–434.
- Wright J., Ma Y., Mairal J., Sapiro G., Huang T. S., Yan S., 2010. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE* 98 (6), 1031–1044.
- Xu H., Zheng J., Chellappa R., 2015. Bridging the domain shift by domain adaptive dictionary learning. In: *British Machine Vision Conference 2015*. Brighton, UK.
- Xu R., Chen Z., Zuo W., Yan J., Lin L., 2018. Deep cocktail network: multi-source unsupervised domain adaptation with category shift. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973.
- Yang M., Zhang L., Feng X., Zhang D., 2011. Fisher discrimination dictionary learning for sparse representation. In: *International Conference on Computer Vision*, pp. 543–550.
- Yang L., Balaji Y., Lim S. N., Shrivastava A., 2020. Curriculum manager for source selection in multi-source domain adaptation. *arXiv preprint. arXiv:2007.01261*.
- Zhao W., Chellappa R., Phillips P. J., Rosenfeld A., 2003. Face recognition: a literature survey. *ACM Computing Surveys (CSUR)* 35 (4), 399–458.

- Zheng J., Liu M. Y., Chellappa R., Phillips J. P., 2012. A Grassmann manifold-based domain adaptation approach. In: Proceedings of the International Conference on Pattern Recognition, pp. 2095–2099.
- Zou Y., Yu Z., Kumar B. V., Wang J., 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 289–305.
- Zou Y., Yu Z., Liu X., Kumar B. V., Wang J., 2019. Confidence regularized self-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5982–5991.

ОБ АВТОРАХ ГЛАВЫ

Йогеш Баладжи – кандидат наук в Мэрилендском университете в Колледж-Парке. Получил степень магистра компьютерных наук в Университете Мэриленда и степень бакалавра технических наук в области электротехники в Индийском технологическом институте в Мадрасе. Его исследовательские интересы связаны с компьютерным зрением и машинным обучением с упором на адаптацию предметной области и генеративное моделирование.

Хиен В. Нгуен – доцент кафедры электроники и вычислительной техники Хьюстонского университета. Он получил кандидатскую степень в Мэрилендском университете в Колледж-Парке (2013 г.) и степень доктора наук в Национальном университете Сингапура (2007 г.). Его научные интересы лежат на стыке машинного обучения и медицины. Опубликовал 50 статей в рецензируемых журналах и является соавтором 12 патентов США. Разработанная им медицинская диагностическая система для помощи врачам была представлена в серии «Великие инновационные идеи» Ассоциации компьютерных исследований. Является заслуженным членом Национальной академии изобретателей.

Рама Челлаппа – почетный профессор на факультетах электроники, вычислительной техники и биомедицинской инженерии Университета Джона Хопкинса (JHU). Занимает постоянную должность профессора Колледж-Парка на факультете ECE в Университете Мэриленда (UMD). До прихода в JHU был заслуженным профессором университета, профессором кафедры ECE и Института перспективных компьютерных исследований Университета Мэриленда в UMD. В настоящее время его научные интересы включают компьютерное зрение, распознавание образов и машинный интеллект. Он получил многочисленные награды за исследования, преподавание, поддержку и наставничество от Университета Южной Калифорнии, Университета Мэриленда, IBM, IEEE, Совета по биометрии IEEE и Международной ассоциации распознавания образов. Ему присвоены звания заслуженного инженера по электронике факультета ECE Университета Пердью и почетного выпускника Индийского института науки. Является членом IEEE, IAPR, OSA, AAAS, ACM, AAAI и NAI и имеет восемь патентов.

Глава 8

Адаптация домена и непрерывное обучение семантической сегментации

Авторы главы:
Умберто Микьели, Марко Тольдо и Пьетро Зануттиг;
кафедра информационной инженерии,
Университет Падуи, Падуя, Италия

Краткое содержание главы:

- формальное представление задачи адаптации домена для семантической сегментации и представление различных уровней, на которых может выполняться адаптация;
- расширенный обзор методов адаптации домена для семантической сегментации;
- обзор последних достижений в области непрерывного обучения для семантической сегментации.

8.1. ВВЕДЕНИЕ

Стандартная методика обучения с учителем предполагает наличие большого обучающего набора, содержащего данные с теми же статистическими свойствами, что и целевые данные, помеченные в соответствии с решаемой задачей. Эта методика легла в основу огромного количества стратегий машинного обучения, от простых линейных классификаторов до продвинутых методов глубокого обучения, и позволила выработать для них надежную теоретическую основу.

Однако при переходе от экспериментов к практическим приложениям пользователи сталкиваются с некоторыми ограничениями. Во-первых, для

обучения модели под прикладную задачу обычно недостает обучающих данных. Несмотря на то что общедоступны очень большие обобщенные наборы данных, сбор и разметка достаточного количества данных для узкой задачи обходятся дорого и отнимают много времени у большинства компаний, разрабатывающих системы машинного обучения. Отсюда вытекает потребность в методах адаптации домена, способных перенести полученные знания из общего исходного набора данных в целевые данные актуальной задачи. Это может быть обучение с частичным привлечением учителя, когда доступно небольшое количество размеченных данных для набора целевого домена, либо обучение без учителя, когда для целевого домена отсутствует разметка или вообще нет данных.

Во-вторых, разметка данных может лишь частично или не совсем точно соответствовать целевой задаче (т. е. так называемая *слабая разметка*, или *слабое обучение*). Многие современные исследования направлены на поиск способов использования разметки данных, относящихся к другой, но связанной задаче, или к набору классов, отличных от целевого. В связи с этим во многих случаях целевая задача не полностью определена в начальной точке, и во время работы модели могут быть добавлены новые классы или даже новые задачи. Для таких сценариев предназначены стратегии *непрерывного обучения* (continuous learning, CL), направленные на постепенное изучение новых задач или классов без переобучения модели машинного обучения с нуля.

Эти соображения применимы ко многим моделям обучения и целевым задачам, но становятся особенно актуальными, когда для обучения требуется огромное количество данных и большие вычислительные усилия. В частности, это касается задач понимания изображений и видео, которые в настоящее время обычно решаются с помощью сложных моделей глубокого обучения. По этой причине в задачах такого рода широко применяется перенос обучения, особенно популярный в области классификации изображений – самой простой классической задаче понимания изображений на глобальном уровне.

В этой главе мы сосредоточимся на более сложной задаче *семантической сегментации*, где, несмотря на классификацию на уровне изображения, выполняется плотная разметка на уровне пикселей. Эта задача является не только более сложной, но и особенно интересной, поскольку операция маркировки занимает очень много времени (намного больше, чем при классификации изображений), что затрудняет создание больших обучающих наборов. Хотя для решения этой задачи можно использовать большое количество подходов, в настоящее время большинство методов используют стратегии глубокого обучения и, в частности, сверточные нейронные сети (CNN) со структурой автокодировщика. Мы тоже возьмем за основу эту архитектуру.

Начнем с задачи *адаптации домена без учителя* (unsupervised domain adaptation, UDA) или с *частичным привлечением учителя* (semisupervised domain adaptation, SDA). Стандартная постановка задачи остается прежней, но происходит смещение статистических свойств данных при переходе от исходного к целевому набору. Мы дадим строгую формулировку задачи в разделе 8.2.1, учитывая также более сложные конфигурации, в которых могут изменяться

и домен, и задача. Адаптация сети глубокого обучения к целевому домену может быть выполнена на разных этапах глубокой сети, т. е. (1) на входном уровне путем трансляции изображений в новый домен, более похожий на целевой, (2) на уровне признаков путем построения пространства признаков, в котором различные классы лучше разделены, что делает описание более устойчивым к изменению домена, и, наконец, (3) на уровне вывода путем обеспечения когерентности между пространствами выходных вероятностей при использовании данных из двух доменов. Для достижения этих целей можно использовать различные стратегии, такие как состязательное обучение, генеративные сети для трансляции доменов, самообучение, минимизация энтропии и многие другие. В разделе 8.2.3 будут подробно представлены наиболее успешные стратегии.

Во второй части этой главы мы проанализируем стратегии непрерывного обучения (continual learning, CL), которые предназначены для реагирования на изменение задач с течением времени. Такие изменения обычно представлены расширением набора меток, добавлением новых меток или разделением существующих на более точные подклассы. Одной из основных целей является возможность адаптировать сеть к новым условиям, используя только данные, касающиеся новых задач, и не переобучая модель с нуля. Однако это весьма нетривиально из-за так называемого явления *катастрофического забывания* (catastrophic forgetting), поскольку модель машинного обучения склонна «забывать» знания о предыдущих задачах при изучении новых. Мы начнем с формального представления этой проблемы в разделе 8.3.1 и рассмотрим широкий спектр экспериментальных сценариев.

Затем в разделе 8.3.3 мы покажем, как можно справиться с проблемой забывания с помощью стратегий сохранения знаний, особенно на основе дистилляции знаний. Другие стратегии основаны на замораживании параметров или замедлении обучения в некоторых частях сети, на попытках регенерировать прошлые (и более недоступные) данные предыдущих классов с использованием генеративных сетей или данных, собранных в интернете.

8.1.1. Формальная постановка задачи

В этом разделе мы сформулируем задачу переноса обучения и введем некоторые обозначения, которые будем использовать в оставшейся части главы. Определим домен $\mathcal{D} = \{\mathcal{X}, P(X)\}$, где \mathcal{X} – пространство входных данных, а $P(X)$ – функция распределения вероятностей по этим входным данным. Задача \mathcal{T} в домене \mathcal{D} представляет собой комбинацию пространства меток \mathcal{Y} с предсказательной функцией $f(\cdot)$, моделирующей условное распределение вероятностей $P(Y|X)$. Таким образом, любую задачу машинного обучения с учителем обычно можно связать с поиском функции $h: \mathcal{X} \rightarrow \mathcal{Y}$, которая лучше аппроксимирует неизвестную $f(\cdot)$ путем изучения набора помеченных обучающих выборок, взятых из совместного распределения $P(X, Y)$ по $\mathcal{X} \times \mathcal{Y}$.

Предположим, что домен входных данных \mathcal{D} не уникален, например существуют отдельно исходный домен \mathcal{D}_S и целевой \mathcal{D}_T (как в классическом UDA) или домен разбит на несколько частей $\mathcal{D}^{(t)}$, $t = 1, \dots, T_{\max}$, доступных

для обучения в разное время (как в классическом CL). Более того, в этих доменах может потребоваться решение разных задач, например для исходного и целевого доменов могут быть выбраны две разные задачи \mathcal{T}_S и \mathcal{T}_T соответственно, или может существовать последовательность задач $\mathcal{T}^{(t)}$, $t = 1, \dots, T_{\max}$, которые нужно решать на разных этапах t процесса обучения. Перенос обучения определяется как поиск улучшенной прогностической функции $f_T(\cdot)$ по целевому домену (или по нескольким целевым доменам), опираясь на полезную информацию, извлеченную из задачи \mathcal{T}_S в исходном домене \mathcal{D}_S , в случае $\mathcal{D}_S \neq \mathcal{D}_T$ или $\mathcal{T}_S \neq \mathcal{T}_T$.

Стоит отметить, что адаптацию домена и непрерывное обучение можно рассматривать как два особых случая переноса обучения; в первом случае исходный и целевой домены разные, а задача одна и та же, а во втором случае макродомен один и тот же (но доступен отдельными частями), а задача меняется.

8.2. АДАПТАЦИЯ ДОМЕНА БЕЗ УЧИТЕЛЯ

За последние несколько лет глубокое обучение оказало огромное новаторское влияние на область компьютерного зрения. До революции глубокого обучения семантическая сегментация считалась очень сложной задачей, и даже сложные алгоритмы обеспечивали лишь посредственную производительность. В настоящее время с появлением глубоких нейронных сетей мы можем получить замечательные результаты при условии наличия приемлемых вычислительных ресурсов. Тем не менее потенциал, заложенный в моделях глубокого обучения, может быть полностью раскрыт только при наличии достаточного объема тщательно размеченных обучающих данных. Сложность, заключенная в миллионах обучаемых параметров современных моделей глубокого обучения, легко приводит к переобучению модели, а не к повышению ее точности, и этому приходится противодействовать, используя огромные наборы данных для обучения. Ярким примером значимости большого объема обучающих данных является крупномасштабный набор данных ImageNet (Deng et al., 2009), чей вклад в раннюю разработку и расширение глубоких нейронных сетей для классификации изображений, безусловно, очень велик.

К сожалению, сбор и аннотирование выборок данных часто обходятся чрезвычайно дорого, отнимают много времени и подвержены ошибкам, поскольку в процессе требуется большое количество интенсивного человеческого труда. Чрезмерная стоимость может помешать сбору достаточного количества данных для решения новой задачи или перехода в новую среду, тем самым препятствуя практическому применению достижений глубокого обучения. Поэтому было бы чрезвычайно полезно использовать ранее созданные наборы данных, если они обладают сходными свойствами с целевыми данными. Уже имеющиеся обучающие выборки могут быть эффективно использованы для решения текущей задачи, если они относятся к домену, коррелирующему с целевым.

Несмотря на то что передача информации из связанных доменов выглядит довольно привлекательной и простой, на практике этот процесс требует осторожности. Глубоким нейронным сетям обычно не хватает навыков обобщения. Другими словами, даже небольшое изменение в распределении данных между обучающими и рабочими статистическими распределениями может привести к серьезному снижению качества вывода модели. По этой причине простое применение предварительно обученной модели в новой среде, скорее всего, потерпит неудачу, поскольку атрибуты, специфичные для домена, обычно фиксируются вместе с атрибутами, не зависящими от домена, что препятствует эффективной передаче знаний. В этом случае пригодится адаптация домена, поскольку она позволяет справиться со статистическим разрывом между исходным и целевым представлениями. Конечной целью усилий по адаптации является обучение прогнозной модели для выбранной задачи, оптимально работающей как в исходном, так и в целевом домене, в то время как обучение в значительной степени (или исключительно) происходит на размеченных данных исходного домена. Следовательно, эффективная передача знаний из исходного домена в целевой имеет решающее значение для достижения в конечном итоге хорошего качества модели. Особенно интересен вариант адаптации домена без учителя (UDA), в котором полностью отсутствует разметка данных целевого домена. Это чрезвычайно благоприятный (но сложный) сценарий, поскольку данные из целевого домена больше не требуют дорогостоящего аннотирования.

В последнее время задача адаптации домена очень активно изучается в контексте глубокого обучения применительно к визуальным задачам. Хотя глубокие сверточные структуры доказали свою способность изучать визуальные признаки, полезные для решения множества связанных задач (например, классификации изображений, обнаружения объектов, семантической сегментации), переносимость этих представлений обычно снижается при переходе на более глубокие сетевые уровни (Long et al., 2015).

Ранние работы по адаптации доменов для глубоких сетей в основном были сосредоточены на задаче классификации изображений. Во многих подходах совместно оценивается и минимизируется послойная мера статистического расхождения доменов, что способствует извлечению представлений признаков, инвариантных к домену, в то время как способность к различению гарантируется функцией потерь для конкретной задачи. Впоследствии чрезвычайно успешными оказались стратегии состязательной адаптации, в схемах которых расхождение доменов выражается посредством обучаемого дискриминатора и минимизация расхождения осуществляется состязательным образом. Более подробная информация о состязательном обучении и его использовании в адаптации домена будет представлена в разделе 8.2.3.1. Этот подход открыл возможности для создания решений по адаптации домена, способных решить задачу семантической сегментации, где более высокая сложность с точки зрения сетевых представлений, необходимых для попиксельной классификации, заставляет использовать более продвинутые приемы.

8.2.1. Формулировка задачи адаптации домена

Адаптация домена (domain adaptation, DA) – это особый случай переноса обучения, так называемое *трансдуктивное трансферное обучение*, в котором исходная и целевая задачи совпадают ($\mathcal{T}_S = \mathcal{T}_T$), тогда как расхождение заключается в различии предметной области ($\mathcal{D}_S \neq \mathcal{D}_T$). Кроме того, подразумевается, что адаптация домена гомогенна, т. е. смещение домена происходит на статистическом уровне ($P(X_S, Y_S) \neq P(X_T, Y_T)$), а не из-за различных входных пространств (\mathcal{X}_S и \mathcal{X}_T принадлежат одинаковой семантической области, например изображения городских сцен) (Wang and Deng, 2018).

В последнее время в некоторых исследованиях рассматривают более сложные сценарии, чем стандартная гомогенная адаптация, позволяющие использовать разные наборы семантических классов в исходной и целевой областях (\mathcal{C}_S и \mathcal{C}_T). В зависимости от того, как связаны исходный \mathcal{C}_S и целевой \mathcal{C}_T наборы, можно выделить несколько сценариев адаптации домена (рис. 8.1):

- *закрытая адаптация*: соответствует гомогенному случаю, когда семантические классы исходного и целевого доменов полностью совпадают ($\mathcal{C}_S = \mathcal{C}_T$);
- *частичная адаптация*: в этом случае существуют некоторые исходные классы, которых нет в целевом домене ($\mathcal{C}_S \supset \mathcal{C}_T$);
- *открытая адаптация*: в отличие от частичной адаптации, здесь допускается наличие некоторых целевых частных классов, для которых отсутствуют обучающие примеры в исходном домене ($\mathcal{C}_S \subset \mathcal{C}_T$);
- *открытая частичная адаптация*: исходный и целевой домены содержат отдельные наборы семантических классов (Saito et al., 2020) с подмножеством общих классов ($\mathcal{C}_S \neq \mathcal{C}_T$, $\mathcal{C}_S \cap \mathcal{C}_T \neq \emptyset$). Однако элементы, принадлежащие подмножеству классов, исключительных для целевого домена, должны быть признаны не относящимися к общим классам;
- *неограниченная адаптация*: этот сценарий очень похож на открытую адаптацию, но объекты целевых частных классов должны быть явно классифицированы, а не только связаны с общим неизвестным целевым классом. Этот вариант был предложен недавно (Bucher et al., 2020) и является самым амбициозным из всех, поскольку допускает полную предварительную неосведомленность о семантическом содержании целевых данных.

В следующих разделах данной главы основное внимание будет уделено стандартной и наиболее распространенной закрытой адаптации, поскольку в настоящее время это, безусловно, наиболее изученный вариант.

В зависимости от степени доступности разметки в целевом домене задача адаптации разделяется на категории, начиная от полностью или частично аннотированных наборов (и, соответственно, полного или частичного обучения с учителем) до полностью лишенного меток набора (обучение без учителя). В частности, мы будем рассматривать адаптацию домена без учителя, поскольку в последнее время наблюдается рост ее популярности, особенно в отношении задачи семантической сегментации, и она охватывает множество практических применений. Предполагается, что доступно множество по-

меченных исходных данных $\{x_i^s, y_i^s\}$, составленное в соответствии с исходным совместным распределением по $\mathcal{X}_S \times \mathcal{Y}_S$, в паре с множеством неразмеченных данных $\{x_i^t\}$, извлеченных из отдельного целевого предельного распределения по \mathcal{X}_T . Цель состоит в том, чтобы найти прогностическую функцию, правильно моделирующую отношение вход–метка задачи в целевой области, в то время как знания о выбранной задаче могут быть извлечены только из исходных размеченных данных.



Рис. 8.1 ❖ Различные варианты адаптации домена в зависимости от того, как связаны наборы исходных и целевых классов

Кроме того, для работы стандартных методов адаптации домена исходный и целевой домены должны быть каким-то образом связаны между собой, т. е. они должны совместно использовать контент, относящийся к задаче, в то время как низкоуровневые атрибуты могут различаться. Этот сценарий обычно называют одноэтапной адаптацией, поскольку передача знаний происходит непосредственно между исходными и целевыми данными без промежуточных этапов.

8.2.2. Основные подходы к адаптации

Как обсуждалось ранее, за ухудшением качества, от которого страдают глубокие прогнозные модели, оказавшиеся в новой целевой среде, лежит явление ковариантного расхождения между распределениями исходных и целевых данных. По этой причине большая часть разработок в области адаптации домена строится на преодолении статистического разрыва между распределениями доменов, чтобы прогнозная модель давала удовлетворительные результаты всякий раз, когда эти распределения совпадают.

Для достижения статистического соответствия были разработаны различные стратегии, которые будут подробно рассмотрены в разделе 8.2.3. Мы

можем разбить эти стратегии на категории в зависимости от того, где в используемой модели семантической сегментации устраняется статистическое несоответствие. В частности, адаптации могут подвергаться различные представления данных, от priоров изображений перед классификацией до промежуточных и выходных активаций сети (рис. 8.2). Далее мы рассмотрим основные идеи, сгруппированные по способу адаптации.

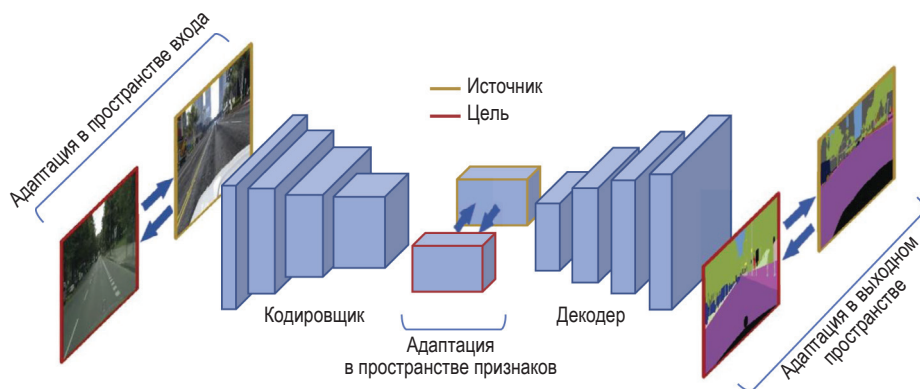


Рис. 8.2 ❖ Обобщенная схема сети автокодировщика для семантической сегментации. Выделены различные этапы сети, на которых могут применяться стратегии адаптации предметной области, от пространства входного изображения до промежуточных или выходных активаций сети

8.2.2.1. Адаптация на входном уровне

Первая стратегия заключается в выполнении адаптации на входе – непосредственно на изображениях до того, как они будут переданы в сеть сегментации (как показано в крайней левой части рис. 8.2). Идея состоит в том, чтобы заставить выборки данных из любой области достичь единообразного внешнего вида, а это означает, что они не только должны обладать семантическим сходством высокого уровня, но также должны быть согласованы их низкоуровневые статистические расхождения. Это связано с тем, что низкоуровневые атрибуты, зависящие от домена, даже если они не определяют семантическое содержание входного изображения, все же могут быть захвачены моделью прогнозирования, что впоследствии приводит к неверным прогнозам, когда атрибуты меняются при смене домена. Ярким примером этого является адаптация модели, обученной на синтетических изображениях, к реальному миру. Хотя синтетические изображения могут выглядеть в высшей степени реалистично, им могут быть присущи специфические черты, пусть даже небольшие, которые могут существенно снизить качество работы модели, обученной на синтетических данных, в реальной среде.

Обычный подход к адаптации домена на входном уровне заключается в сопоставлении данных с новым пространством изображения, где спроецированные исходные (или целевые) изображения несут улучшенное percep-

тивное сходство с целевыми (или исходными). Как правило, это достигается с помощью методов переноса стиля, которые работают путем сопоставления исходного и целевого предельных распределений в пространстве изображения. Подавая данные из нового доменно-инвариантного пространства в сеть сегментации, предсказатель теперь должен иметь возможность сохранять согласованные результаты независимо от доменов.

Преимуществом этого метода является его полная независимость по отношению к используемой сети сегментации, которая не нуждается в каких-либо модификациях. За это удобство, однако, приходится платить тем, что в стандартной схеме без каких-либо дополнительных упорядочивающих факторов маргинальное выравнивание может быть выполнено без одновременного сопоставления распределений, обусловленных классом. Другими словами, иногда можно получить инвариантные представления домена, которым все же не хватает семантической согласованности с исходными данными, которая имеет ключевое значение для решения задачи сегментации. Чтобы обойти эту проблему, было предложено несколько решений для достижения семантически согласованных переводов изображений, например с помощью ограничений реконструкции изображения или дополнительных компонентов потерь, обеспечивающих согласованность прогнозов сегментации.

8.2.2.2. Адаптация на уровне признаков

Альтернативный подход состоит в том, чтобы сосредоточить адаптацию на представлениях признаков, стремясь к выравниванию распределения скрытых представлений сети, которые обычно извлекаются из выходных данных кодировщика в традиционной архитектуре автокодировщика (даже если применялась адаптация на других этапах сети). В данном случае мы стремимся построить инвариантное к домену скрытое пространство, в котором признаки, извлеченные либо из исходных, либо из целевых входных изображений, имеют одно и то же распределение. В конце концов, обучение исключительно на исходных представлениях должно привести к хорошей точности также и в целевой области, поскольку общая классификация в адаптированном скрытом пространстве должна быть одинаково точна как для исходных, так и для целевых представлений (при их одинаковом распределении).

В контексте семантической сегментации пространство признаков сохраняет значительную сложность из-за своей высокой размерности, которая необходима для того, чтобы прогнозная модель собирала глобальные семантические подсказки, одновременно достигая точности на уровне пикселей. Кроме того, что касается адаптации на входном уровне, семантически независимое выравнивание частных распределений (например, стандартная состязательная адаптация) не гарантирует, что совместные распределения вход-метка совпадают, поскольку из неразмеченных целевых выборок нельзя получить никакой информации о целевом совместном распределении. По этим причинам многие методы адаптации на уровне признаков, которые были успешно разработаны для классификации изображений, с трудом распространяются на задачу плотной сегментации и, как правило, требуют тщательной настройки и дальнейшей регуляризации.

8.2.2.3. Адаптация на уровне выхода

Наконец, последний класс методов адаптации домена использует выравнивание распределений между доменами по выходным данным сети, т. е., как правило, по выходному вероятностному пространству для каждого класса. Доказано, что карты прогнозной вероятности не только сохраняют достаточную сложность и богатство семантической информации, но также охватывают низкоразмерное пространство, в котором статистическое выравнивание достигается гораздо более эффективно, например с помощью состязательной стратегии. Кроме того, исходное знание можно косвенно транслировать в немаркированный целевой домен, прибегая к той или иной форме самостоятельного обучения. Доказано, что приоры, полученные из распределения меток исходного домена, также обеспечивают полезную регуляризацию процесса обучения, поскольку они обычно определяют высокоуровневые семантические свойства, общие для разных доменов.

8.2.3. Методы адаптации домена без учителя

Далее мы обсудим наиболее эффективные подходы к адаптации домена в семантической сегментации с обучением без учителя. Для каждого набора методов мы приведем несколько исследовательских работ, которые имеют отношение к этой категории и могут быть полезны читателям. Однако следует подчеркнуть, что большинство недавно предложенных стратегий адаптации домена используют комбинацию нескольких методов для повышения качества.

8.2.3.1. Состязательная адаптация домена

Первоначально состязательное обучение предназначалось для создания изображений (Goodfellow et al., 2014). Основная цель генеративной задачи – извлечь неизвестные данные моделируемого распределения вероятностей из используемого обучающего набора. Состязательная стратегия оказалась чрезвычайно эффективной при решении этой задачи, поскольку не требуется находить явное выражение целевого распределения данных и, что более важно, для обучения генеративной модели не требуется придерживаться какой-либо цели. Процесс обучения основан на минимаксной игре, в которой сеть генератора в состязании с сетью дискриминатора учится создавать реалистичные образы. Дискриминатор – это бинарный классификатор, целью которого является различение исходных обучающих данных и данных, созданных генератором. Генератор представляет собой генеративную модель, которая принимает на вход случайный шум (или некоторые обуславливающие данные в более поздних вариантах) и создает данные (т. е. изображения в интересующей области), напоминающие те, что есть в обучающем наборе. Генератор стремится постоянно повышать реалистичность создаваемых изображений, чтобы обмануть дискриминатор, и это достигается с помощью функции потерь, минимизация которой, в свою очередь, максимизирует ошибки дискриминатора. Модель обучается путем чередования

этапа обучения дискриминатора с целью увеличения точности различения и этапа оптимизации генератора с противоположной целью (см. рис. 8.3). Правильно выполненное состязательное обучение должно привести к статистическому распределению сгенерированных данных, полностью совпадающему с обучающим набором, а это означает, что исходные и сгенерированные данные должны быть статистически неразличимы. Кроме того, дискриминатор должен быть в состоянии как извлекать, так и выражать меру статистического несоответствия в форме структурированной потери обучения. Следовательно, целевую функцию можно рассматривать как совместно изученную и оптимизированную в состязательном процессе, что позволяет ей адаптироваться к конкретному контексту.

Состязательное обучение было успешно распространено на задачу адаптации домена. Дискриминатор теперь превращается в классификатор домена, который используется для управления процессом адаптации. Его различительная способность фактически направлена на улавливание статистического несоответствия между представлениями из разных доменов, которое отвечает за снижение точности и, следовательно, должно быть минимизировано.

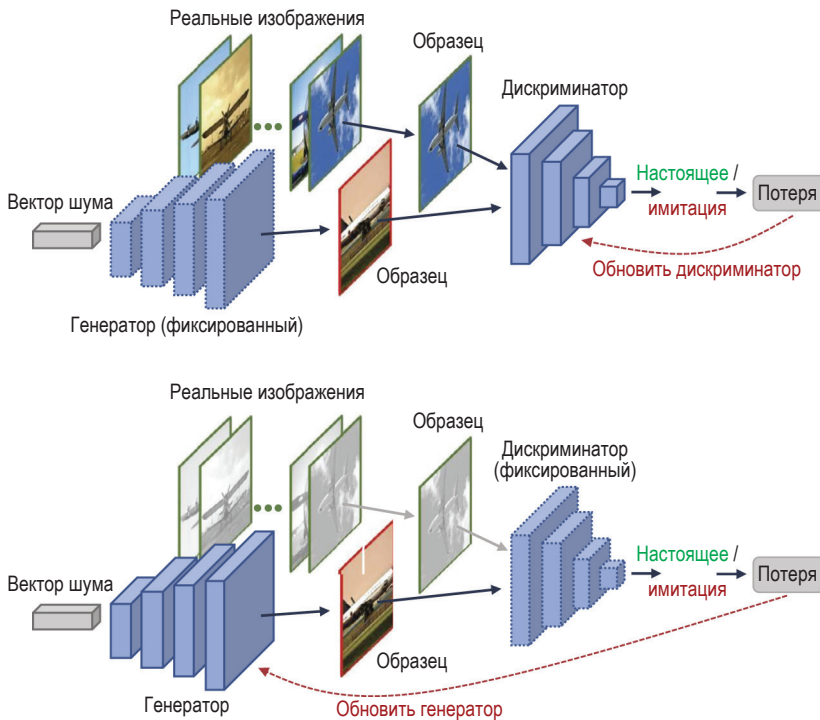


Рис. 8.3 ❖ Обучение генеративно-состязательной сети.
Этап обновления дискриминатора (вверху) и генератора (внизу)

У классификатора домена есть два возможных применения. Во-первых, он может различать внутренние и выходные представления, извлеченные из данных в исходном или целевом домене (рис. 8.4). Это позволяет вводить

дополнительные члены потерь, обеспечивающие конструирование признаков или выходных пространств, которые более инвариантны к домену. Во-вторых, можно использовать дискриминатор, чтобы различать выходные данные сети (которые могут соответствовать входным данным из исходного или целевого домена) и эталонные сегментации (которые при обучении без учителя присутствуют только в исходном домене). Поскольку в состязательной модели нет необходимости иметь эталонные данные, соответствующие предоставленным изображениям, это позволяет также использовать изображения целевого домена, для которых нет эталонов, и добиться того, чтобы их предсказанные карты сегментации имели статистические свойства, аналогичные эталонным (рис. 8.5). В соответствии с этой стратегией к стандартному управляющему сигналу от аннотированных исходных данных присоединяется управляющий сигнал от дискриминатора домена, который подталкивает сеть к инвариантности домена, что, в свою очередь, уменьшает внутреннее расхождение доменов в направлении исходного домена.

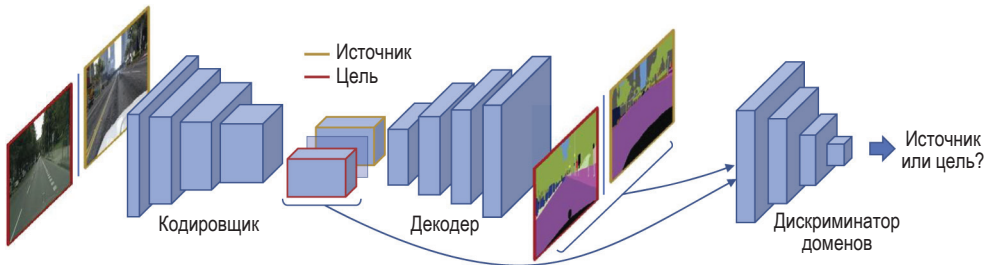


Рис. 8.4 ❖ Иллюстрация стандартной стратегии состязательной адаптации. Дискриминатор домена отмечает статистическое несоответствие между исходным и целевым представлениями (например, выходными данными сети сегментации или картами признаков, вычисленными из того или иного домена). Затем его обучающий сигнал используется для сближения доменов

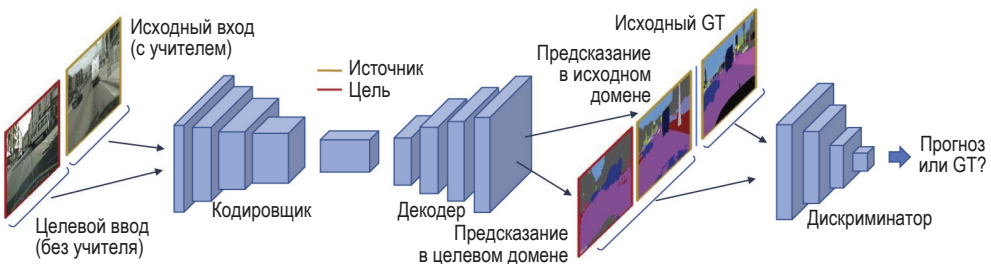


Рис. 8.5 ❖ Иллюстрация стратегии состязательной адаптации на уровне выхода, где сближение доменов выполняется косвенно путем устранения разрыва в распределении между исходными картами аннотаций и сетевыми прогнозами из исходного или целевого домена

После успеха состязательной адаптации домена для классификации изображений (Ganin, Lempitsky, 2015; Ganin et al., 2016) состязательная

стратегия была применена также в контексте семантической сегментации для достижения сближения доменов по представлению скрытых признаков (Hoffman et al., 2016). Тем не менее, как упоминалось ранее, глобальное сближение маргинальных распределений доменов, обеспечиваемое состязательной схемой, может привести к неправильной передаче семантических знаний между доменами, когда в процессе обучения игнорируются распределения, обусловленные классом. По этой причине для достижения эффективной состязательной адаптации при решении задачи семантической сегментации в конвейер адаптации необходимо встроить дополнительные модули.

Возможное решение – интегрировать состязательное выравнивание признаков в генеративный подход (раздел 8.2.3.2), как это сделано в нескольких работах (Li et al., 2019; Hoffman et al., 2018; Chen et al., 2019; Toldo et al., 2020). Цель данного подхода заключается в усилении адаптации пространства изображения таким образом, чтобы перенос атрибута, направленный на совмещение визуальных представлений изображений из разных доменов, был расширен внутри пространства признаков. Альтернативой является адаптация по категориям (Chen et al., 2017; Du et al., 2019). Идея состоит в том, чтобы прибегнуть к состязательному обучению по классам, вводя несколько дискриминаторов отдельных признаков для каждого класса, что, в принципе, должно обеспечить семантически непротиворечивую передачу знаний, которая отсутствует в стандартной глобальной адаптации. Наконец, зайдя с другой стороны, можно прибегнуть к ограничению реконструкции для обеспечения инвариантности домена относительно скрытых представлений признаков (Sankaranarayanan et al., 2018; Murez et al., 2018; Zhu et al., 2018). В этом случае состязательное обучение применяется к пространству реконструируемого изображения, чтобы гарантировать, что представления признаков могут быть спроецированы обратно либо в исходное, либо в целевое пространство изображений без каких-либо различий.

Как отмечалось ранее, решение задачи семантической сегментации сопровождается построением довольно сложного пространства признаков из-за высокой размерности представлений. Чтобы обойти сложность, связанную с адаптацией пространства признаков, в некоторых исследованиях была предложена адаптация в выходном пространстве сегментации (Tsai et al., 2018; Chen et al., 2018; Chang et al., 2019; Luo et al., 2019; Yang et al., 2020; Biasetton et al., 2019; Michieli et al., 2020; Spadotto et al., 2020). И действительно, было показано, что низкоразмерные выходные представления сохраняют достаточно семантической информации для успешной адаптации. В этой новой состязательной схеме выходного уровня дискриминатор домена учится распознавать домен, из которого исходят карты сегментации. В то же время сеть сегментации играет роль генератора, предоставляя междоменные статистически близкие прогнозы, чтобы обмануть классификатор доменов. Хотя сближение исходных и целевых выходных представлений стало общепризнанным решением (Tsai et al., 2018; Chen et al., 2018; Chang et al., 2019; Luo et al., 2019; Yang et al., 2020), в некоторых последующих работах стандартный подход был пересмотрен путем поиска косвенного сближения доменов (Biasetton et al., 2019; Michieli et al., 2020; Spadotto et al., 2020), когда прогнозы из

того и другого доменов заставляют распространяться как эталонные метки источника (как показано на рис. 8.5).

Некоторые новые подходы (Vu et al., 2019; Tsai et al., 2019) были основаны на извлечении значимых паттернов из выходного пространства сегментации. Идея состоит в том, чтобы предоставить дискриминатору домена более функциональное и значимое понимание исходных и целевых представлений, что позволит ему вырабатывать более эффективный управляющий сигнал в процессе адаптации. Это делается путем ручного извлечения некоторой значимой информации как из исходных, так и из целевых данных (например, карты энтропии по прогнозам сегментации (Vu et al., 2019)) для подачи в сеть дискриминатора. В свою очередь, сосредоточив внимание на значимых семантических подсказках из представлений данных, можно усилить состязательное выравнивание по активациям сети сегментации.

8.2.3.2. Генеративная адаптация

Преобразование изображения в изображение – это класс генеративных методов, основной целью которых является построение подходящей функции для проецирования изображений из одного домена в другой. Другими словами, идея состоит в том, чтобы обнаружить совместное распределение данных изображения из разных доменов. Задача преобразования изображения в изображение может быть эффективно использована при адаптации домена. По сути, мы добиваемся переноса визуальных атрибутов из целевого домена в исходный с сохранением исходной семантической информации. Таким образом смягчается явление ковариантного смещения, вызывающее снижение качества классификатора. Двигаясь в этом направлении, некоторые исследователи использовали стратегию адаптации на уровне ввода, основанную на трансляции изображения между исходным и целевым доменами. Общим для всех этих работ является поиск предметной инвариантности внешнего вида изображений из разных доменов. В конечном итоге это позволяет использовать для обучения с учителем транслированные, но все еще аннотированные исходные изображения.

Общий подход, использованный во множестве работ (Hoffman et al., 2018; Chen et al., 2019; Toldo et al., 2020; Zhou et al., 2020; Li et al., 2019; Murez et al., 2018; Qin et al., 2019; Li et al., 2018; Yang et al., 2020b; Gong et al., 2019), заключается в применении успешной модели CycleGAN (Zhu et al., 2017) для выполнения перевода изображения в изображение без учителя (рис. 8.6). Данный фреймворк параллельно состязательным образом изучает условные трансляции изображений как в направлении от источника к цели, так и в обратном направлении. Два состязательных генеративных модуля дополнительно связаны ограничением согласованности цикла, заставляющим каждый из них изучать обратную проекцию другого. Требованием к реконструкции является сохранение геометрии и компоновки входной сцены, но, в свою очередь, не гарантируется сохранение семантического содержания входного изображения при трансляции.

С отсутствием семантической согласованности в стандартной схеме трансляции борется подход с использованием семантического предиктора в сети

сегментации (Hoffman et al., 2018; Chen et al., 2019; Toldo et al., 2020; Zhou et al., 2020; Li et al., 2019). В частности, семантический предиктор можно использовать для обнаружения и, таким образом, предотвращения любого возмущения семантического вывода, которое может произойти во время трансляции, выполняемой генераторами CycleGAN. Как правило, это достигается путем применения согласованных карт прогнозирования к исходным и транслированным версиям одного и того же изображения. Альтернативным решением могло бы быть достижение семантической обоснованности адаптации на основе перевода изображения в изображение, за счет прямого воздействия на состязательные модули перевода. Этот подход, например, был реализован с использованием мягкой потери, чувствительной к градиенту (Li et al., 2018), и ограничения фазовой согласованности (Yang et al., 2020), которые обеспечивают эффект регуляризации по сравнению со стандартным состязательным обучением. Наконец, можно попробовать применить трансляцию из целевого домена в исходный. Доказано, что это уменьшает смещение в сторону исходного домена (Yang et al., 2020). В отличие от стандартной генеративной адаптации, инвариантность домена на уровне изображения в этом случае достигается внутри исходного домена, где псевдоразметка позволяет использовать схожие с источником транслированные целевые изображения для дискриминации с обучением.

В качестве альтернативы адаптации на основе CycleGAN также были изучены методы переноса стиля для достижения доменной инвариантности низкоуровневых атрибутов изображения. В основе этих методов лежит принцип, согласно которому любое изображение можно разделить на его контент и стиль. В то время как стиль изображения связан с низкоуровневыми специфичными для домена признаками, контент указывает на высокоуровневые семантические свойства, не зависящие от домена. Следовательно, объединение исходного контента с целевым стилем должно обеспечивать целевые обучающие данные с сохранением исходных семантических аннотаций. Благодаря генерации новых изображений целевого домена с сохранением разметки может быть выполнена *дискриминация с обучением*¹. Распространенным подходом к переносу стиля является использование декомпозиции контента и стиля в скрытом пространстве (Chang et al., 2019; Pizzati et al., 2020). Последующая трансляция из исходного в целевой домен сводится к задаче объединения извлеченных представлений исходного контента со случайными целевыми стилями, при этом смешанные представления должны быть перепроецированы в пространство изображения. Чтобы избежать сложности, связанной с созданием изображений GAN (особенно сложно получить изображения с высоким разрешением), также были изучены различные типы методов передачи стиля, начиная от нейронной или фотореалистичной передачи стиля (Zhang et al., 2018; Dundar et al., 2018) и до повторной нормализации признаков (Choi et al., 2019; Wu et al., 2019) и низкоуровневого управления частотным спектром (Yang, Soatto, 2020).

¹ При дискриминации с обучением на вход дискриминатора поступают обучающие образцы, категориальная принадлежность которых известна. – *Прим. перев.*

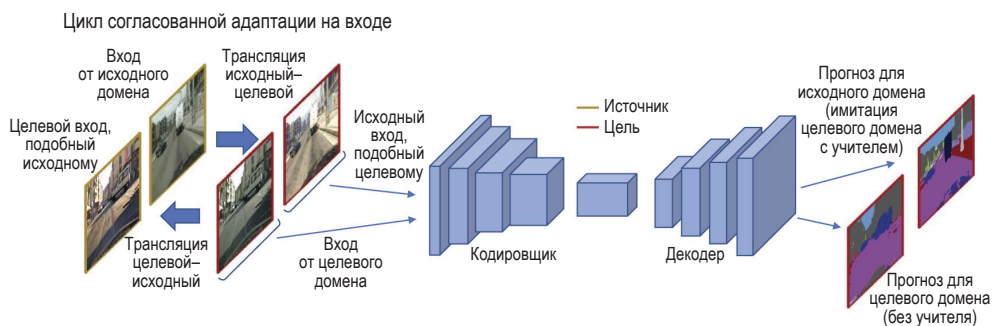


Рис. 8.6 ❖ Схема генеративного подхода к адаптации, основанного на преобразовании изображения в изображение. В частности, исходные входные изображения после трансляции используются как разновидность искусственных данных для дискриминации с обучением

8.2.3.3. Несоответствие классификатора

Как упоминалось в разделе 8.2.3.1, состязательная адаптация на уровне признаков в ее стандартной реализации включает дополнительный классификатор домена, чья способность разделять признаки из исходного и целевого доменов обеспечивает эффективный управляющий сигнал для процесса обучения, подталкивая сеть сегментации к инвариантности домена в скрытом пространстве, которое он охватывает. В свою очередь, механизм отдельных ориентированных на задачи целей отвечает за то, чтобы предсказательная сеть изучила реальную задачу с обучающим источником, т. е. применила стандартную кросс-энтропийную потерю для семантической сегментации.

Несмотря на то что стандартная состязательная адаптация довольно эффективна, ей не хватает семантической осведомленности (Saito et al., 2018). Надлежащее состязательное сближение, по сути, влечет за собой совмещение маргинальных распределений исходных и целевых данных, за которым в общем случае не следует статистическое совмещение по условным классам. Это связано с тем, что совместные распределения на уровне категорий обязательно остаются неизвестными для классификатора домена, поскольку полное отсутствие обучения с учителем в целевой предметной области подразумевает отсутствие информации о семантическом содержании целевых данных. В результате признаки могут перемещаться вблизи границ классов, где неопределенность классификации может привести к неправильным прогнозам. Хуже того, не зависящий от класса перенос целевых признаков может неправильно совместить их с исходными представлениями другого семантического класса в скрытом пространстве, инвариантном к предметной области, что означает наличие так называемого *отрицательного переноса*.

Стремясь решить эти проблемы, некоторые исследователи (Saito et al., 2018) полностью переработали исходный состязательный подход к адаптации домена. В частности, они отводят роль дискриминатора плотному классификатору для конкретной задачи (т. е. сети кодировщика), которая ранее была назначена дискриминатору внешнего домена. Возмущающая классифика-

тор с помощью отсева, можно определить, где прогнозы более неопределенны, что сильно связано с расстоянием представлений признаков от границ решения. В этой новой состязательной схеме плотный классификатор обучается повышать свою чувствительность к семантическим вариациям целевых представлений. Состязаясь с ним, экстрактор признаков (т. е. кодировщик) стремится обеспечить высокую категориальную достоверность вычисляемых им целевых признаков. Этот механизм должен эффективно удалить заключенную в целевых представлениях, но не связанную с задачей информацию, которая является причиной сильной изменчивости прогнозов, в конечном итоге отодвигая их далеко от границ принятия решений.

Недостатком стратегии несоответствия классификатора в ее исходной форме является присущая декодеру чувствительность к шуму (Saito et al., 2018), которая имеет решающее значение для определения близости целевых выборок к границам классификации, но снижает точность всей сети сегментации, требуя, по сути, дополнительного этапа обучения, чтобы правильно изучить задачу сегментации. В этом отношении первоначальную схему можно было бы улучшить, заменив стратегию отсева для извлечения нескольких прогнозов по одному и тому же представлению признаков парой различных декодеров, которые одновременно обучены обеспечивать правильные, но четкие, плотные классификации (Saito et al., 2018). Это должно эффективно повысить точность секции декодера прогнозной модели за счет дополнительного модуля, который также необходимо обучить во время основного процесса обучения.

Согласно другому подходу, исходную схему состязательного обучения можно изменить, выбрав нестохастический механизм виртуального отсева для поиска масок отсева на минимальном расстоянии, вызывающих максимальное расхождение прогнозов (Lee et al., 2019). При этом исходное решение на основе отсева для получения четких прогнозов от одного классификатора сохраняется, в то время как одновременно решается вышеупомянутая проблема восприимчивости к шуму.

В работе (Luo et al., 2019) дополнительно был исследован принцип совместного обучения, основанный на нескольких предикторах для оценки текущего качества адаптации. В частности, можно использовать обнаружение противоречивых прогнозов, чтобы сосредоточить усилия дискриминатора (теперь в стандартной состязательной структуре предметной области) на менее адаптированных областях входного изображения, т. е. на участках с наибольшей неопределенностью. В конечном итоге это привело к более эффективному сближению доменов исходного и целевого представлений.

8.2.3.4. Самостоятельное обучение

Из-за сходства между двумя задачами несколько методов адаптации домена без учителя (unsupervised domain adaptation, UDA) были заимствованы из области *обучения с частичным участием учителя* (semi-supervised learning, SSL). Действительно, UDA можно рассматривать как крайнюю форму SSL, поскольку в обоих случаях часть обучающих данных не помечена. Однако к исходному отсутствию аннотаций SSL в неразмеченном обучающем наборе

добавляется статистическое расхождение в UDA между исходными и целевыми данными, что требует дополнительных усилий для устранения.

Самонастройка. В последние годы разработан целый класс методов адаптации (Zou et al., 2018, 2019; Li et al., 2019; Biasetton et al., 2019; Michieli et al., 2020; Spadotto et al., 2020; Choi et al. al., 2019; Chen et al., 2019a; Yang, Soatto, 2020; Zhou et al., 2020), использующих самонастройку (self-training). Этот подход, обычно принятый в SSL (Grandvalet, Bengio, 2005), работает за счет создания псевдометок на основе высоконадежных сетевых прогнозов, сделанных на основе немаркированных целевых данных. Таким образом, в целевом домене доступна форма самообучаемого контроля, которую можно использовать в сочетании со стандартным контролем на основе исходных помеченных данных.

В отличие от других методов адаптации, описанных в предыдущих разделах, таких как наиболее успешные состязательные подходы, сближение доменов на уровне признаков неявно осуществляется посредством целевого самообучения, поскольку исходные обучающие данные косвенно переносятся на целевой домен с помощью псевдометок. Самонастройка подталкивает выходные данные вероятности сети к пиковому распределению, что позволяет делать прогнозы, демонстрирующие более уверенное поведение. Ключевая проблема, однако, заключается в самореферентности этого метода, что может привести к катастрофическому распространению ошибок, если их не обрабатывать должным образом. Излишне самоуверенные неверные прогнозы в неопределенной области фактически могут остаться незамеченными, поскольку отсутствует какой-либо обучающий надзор за неразмеченными данными целевого домена. В свою очередь, эти ошибки прогнозирования могут быть подкреплены стратегией самообучения, вызывая постепенно нарастающее отклонение от правильного решения. Чтобы справиться с этой проблемой, большинство подходов к адаптации, основанных на самообучении, применяют к процессу псевдомаркировки определенные стратегии фильтрации, так что ошибки прогнозирования, по своей сути влияющие на карты целевой сегментации, в значительной степени отбрасываются.

Распространенный подход к адаптации на основе самонастройки включает автономные методы вычисления псевдометок (Zou et al., 2018, 2019; Li et al., 2019), при этом в процессе адаптации доверительный порог обновляется несколько раз путем просмотра всего доступного обучающего набора. В частности, используется итеративная процедура оптимизации самонастройки, которая чередует этапы самонастройки и обучения с учителем на искусственных аннотациях как исходного, так и целевого доменов. В течение всего этапа настройки искусственные аннотации целевого домена остаются фиксированными. Эта автономная стратегия обеспечивает стабильный процесс обучения за счет дополнительной вычислительной нагрузки из-за нескольких этапов псевдоаннотирования всего целевого набора данных.

В другом подходе стратегия самонастройки может быть связана с состязательной адаптацией на уровне выхода (Biasetton et al., 2019; Michieli et al., 2020; Spadotto et al., 2020). В частности, выходную карту из полностью сверточного дискриминатора домена на уровне выхода можно рассматривать как точную меру надежности предсказания данных целевого домена,

таким образом получая полезную информацию для уточнения целевых псевдометок. Итак, качество искусственных аннотаций постепенно улучшается на протяжении всего процесса обучения, поскольку они вычисляются для отдельных пакетов целевых изображений, а не для всего набора данных, что в целом приводит к довольно эффективной адаптации.

В качестве альтернативы надежность псевдометок можно повысить, прибегнув к ансамблям прогнозов (Choi et al., 2019; Chen et al., 2019; Yang, Soatto, 2020; Zhou et al., 2020). Например, можно использовать дополнительную сеть для самонастройки на неразмеченных образцах (Чой и др., 2019; Чжоу и др., 2020). Это делается путем введения сети учителя в дополнение к исходной, которая играет роль ученика. Затем модель-учитель, чьи веса усредняются по весам учеников на прошлых этапах настройки, используется для управления процессом обучения сети-ученика, давая целевые прогнозы, которым ученик вынужден подражать. Наставничество сети-учителя приводит к более точным прогнозам целей, для которых может выполняться менее шумная псевдомаркировка. В результате получается более эффективная адаптация с самонастройкой.

Минимизация энтропии. Минимизация энтропии, также позаимствованная из области обучения с частичным участием учителя, недавно была введена в UDA (Vu et al., 2019). Идея, стоящая за этим подходом, заключается в том, что исходные прогнозы более склонны демонстрировать уверенное поведение, что проявляется в низком уровне энтропии вероятностных выходных данных. И наоборот, выходные карты сегментации из входных данных целевого домена, вероятно, будут демонстрировать большую неопределенность (высокую энтропию), при этом шумовая картина широко размыта и не ограничивается только областями, близкими к семантическим границам. Таким образом, отражая чрезмерно самоуверенное поведение источника в неопределенном целевом домене, сеть сегментации должна, в принципе, преодолеть разрыв в качестве, существующий между доменами. Точнее, эффект минимизации энтропии заключается в том, чтобы избежать пересечения границами классификации областей с высокой плотностью в скрытом пространстве, в то время как целевые представления хорошо сгруппированы вдали от этих границ.

Первоначальная стратегия минимизации энтропии (Vu et al., 2019) работает на уровне пикселей, при этом каждая отдельная пространственная единица независимо способствует достижению общей цели. Однако для преодоления некоторых присущих этому подходу ограничений были введены дополнительные механизмы (Vu et al., 2019; Chen et al., 2019a; Yang, Soatto, 2020). Возможное решение состоит в том, чтобы добиться глобального сближения распределения энтропии с помощью составительного подхода к адаптации домена, где дискриминатор домена получает карты энтропии, а не напрямую выходные вероятностные диаграммы, как в стандартной схеме из раздела 8.2.3.1 (Vu et al., 2019). Следовательно, в этом методе используется структурная информация, заключенная в картах энтропии, что приводит к более эффективной статистической адаптации домена. Кроме того, следует отметить, что цель минимизации энтропии в ее первоначальном виде (Vu et al., 2019) приводит к быстрому градиентному взрыву при переходе от

областей высокой неопределенности к области низкой, что может серьезно затруднить процесс обучения. Решением данной проблемы может стать изменение стандартной цели (например, на основе квадратичных потерь), на цель, аналогичную исходной, но с улучшенными свойствами градиентного сигнала (Chen et al., 2019). Эта стратегия вместе с категориальными весовыми коэффициентами для балансировки вклада различных семантических классов значительно улучшает процесс адаптации.

В нескольких недавних работах минимизация энтропии применялась вместе с методами формирования пространства признаков (Toldo et al., 2021; Barbato et al., 2021). В соответствии с этим подходом, помимо использования минимизации энтропии, заставляют представления внутренних признаков быть сгруппированными, разреженными и ортогональными (если они принадлежат разным классам) как в исходном, так и в целевом доменах, чтобы улучшить адаптацию на уровне признаков. Другая недавняя работа (Barbato et al., 2021) дополнительно вводит ограничение сближения нормы, чтобы помочь цели ортогональности признаков по классам распространять непересекающиеся наборы каналов активных признаков между отдельными семантическими категориями, при этом направляя целевые вложения к высоконадежному (т. е. связанному с высокими значениями нормы признака) исходному распределению.

8.2.3.5. Многозадачность

Последний класс методов адаптации, который необходимо обсудить, относится к многозадачности (Lee et al., 2019; Vu et al., 2019; Chen et al., 2019; Watanabe et al., 2018). Регуляризация достигается путем решения нескольких связанных задач (например, многие подходы сосредоточены на глубинной регрессии) в дополнение к задаче семантической сегментации. Цель состоит в том, чтобы неявно извлечь инвариантные предметные области и семантически значимые представления из изображений, поскольку они должны быть более подходящими для одновременного решения связанных задач. *Глубинная регрессия* (depth regression) обычно используется в сочетании с семантической сегментацией, чтобы упорядочить процесс адаптации на входном уровне на основе преобразования исходного изображения в целевое (раздел 8.2.3.2). Также было показано, что многозадачная адаптация эффективно интегрируется с другими подходами к адаптации. Например, ее можно использовать для улучшения стратегии максимального классификатора (раздел 8.2.3.3), где вместо одного плотного классификатора используются два отдельных модуля декодера для получения карт глубины и карт сегментации (Watanabe et al., 2018). Можно сочетать многозадачность с методом минимизации состязательной энтропии (раздел 8.2.3.4) (Vu et al., 2019), объединяя карты собственной информации с картами прогнозирования глубины перед их подачей на дискриминатор домена. Таким образом повышается способность обнаружения доменным дискриминатором расхождения представлений исходного и целевого доменов, что в конечном итоге обеспечивает более надежное статистическое сближение.

8.3. НЕПРЕРЫВНОЕ ОБУЧЕНИЕ

В последнее время методы глубокого обучения особенно быстро развивались в направлениях, которые ранее считались чрезвычайно сложными, в частности в области компьютерного зрения, где модели глубокого обучения во многих случаях достигают человеческого уровня точности. Методы глубокого обучения постепенно совершенствовались, и переход от академических исследований к различным практическим и промышленным приложениям был лишь вопросом времени. Однако с началом практического применения моделей возникла потребность в методах, позволяющих со временем вводить новые знания, чтобы выполнять новые задачи, не забывая при этом предыдущие. В этом и заключается парадигма непрерывного обучения. Другими словами, когда модели глубокого обучения внедряются в реальный мир, нам нужно иметь возможность улучшать возможности моделей с помощью нового опыта или меток, не переобучая их с нуля.

В целом основная проблема вычислительных моделей заключается в том, что они склонны к катастрофическому забыванию (McClelland et al., 1995; McCloskey, Cohen, 1989), т. е. обучение модели на новых данных мешает использованию ранее изученных знаний и, как правило, сильно ухудшает их представление. Модели глубокого обучения, в частности, предполагают, что на этапе обучения доступны все выборки данных, и поэтому при адаптации к изменению данных они требуют обучения на полном наборе. При обучении на последовательных задачах с ограниченными выборками, поступающими с течением времени, качество работы модели с ранее изученными задачами значительно снижается, поскольку параметры сети оптимизируются для новой задачи без учета старых, если не используются специальные условия (Kemker et al., 2018; Parisi et al., 2019).

Непрерывное обучение (continual learning, CL, также называемое постепенным обучением, обучением на протяжении жизни модели или бесконечным обучением) представляет собой набор методов, разработанных для использования в сложной ситуации, когда последовательно выполняются меняющиеся задачи. Несмотря на то что эта проблема вычислительных моделей давно известна (McCloskey and Cohen, 1989), первые успехи в глубоком обучении появились относительно недавно.

Чтобы еще лучше осознать актуальность проблемы, стоит рассмотреть аналогию между обучением машины и человека. Люди сталкиваются с непрерывным потоком обучающих данных и способны обобщать схожие неявные задачи (Thrun, Pratt, 2012). На протяжении долгих лет ученые предпринимали неоднократные попытки понять механизмы обучения мозга и перенести их на вычислительные модели (Grossberg, 2013; Ditzler et al., 2015).

В последнее время эту проблему активно изучают применительно к некоторым задачам с разметкой на уровне изображения (т. е. с одной или несколькими метками для каждого изображения), такой как классификация изображений (Rebuffi et al., 2017; Li, Hoiem, 2017; Castro et al., 2018) и обнаружение объектов (Shmelkov et al., 2017). В задачах плотной разметки, таких как семантическая сегментация, из-за присущей им повышенной сложности

к решению этой проблемы приступили совсем недавно (Ozdemir, Goksel, 2019; Tasar et al., 2019; Michieli and Zanuttigh, 2019, 2021b; Cermelli et al., 2020; Klingner et al., 2020; Douillard et al., 2021).

Прежде чем углубиться в определение непрерывного обучения и изучить его применение к задачам плотной разметки, мы хотели бы отдельно отметить некоторые общие обзоры по теме, не имеющие прямого отношения к семантической сегментации. Можно выделить работу, посвященную катастрофическому забыванию в сетевых моделях (French, 1999), в то время как статья (Parisi et al., 2019) – это первый обзор, в котором критически сопоставляются недавние работы об этом явлении в моделях глубокого обучения. В (De Lange et al., 2019) многие подходы сравниваются в общем плане, а в работе (Lesort et al., 2020) рассмотрены проблемы непрерывного обучения с особым акцентом на робототехнику.

8.3.1. Формулировка задачи непрерывного обучения

Непрерывное обучение (CL) можно рассматривать как частный случай обучения с переносом, когда распределение домена меняется на каждом шаге приращения и модель должна хорошо работать на всех распределениях. Этот сценарий также тесно связан с задачей адаптации домена, но в данном случае основное внимание уделяется как входным данным, так и аннотациям, распределение которых со временем меняется, и их количество также может быть увеличено (т. е. понадобится различать больше классов).

С другой стороны, стоит отметить, что обсуждение адаптации домена в предыдущем разделе было сфокусировано на входных распределениях домена, где обычно и выполняется адаптация одного домена. В последнее время появились гибридные подходы, сочетающие UDA и CL для преодоления множественных изменений в доменах и задачах (Busto, Gall, 2017; Zhuo et al., 2019; Kundu et al., 2020).

Из-за присущего непрерывному обучению разнообразия проблем и связанных с ними трудностей в большинстве подходов ослабляют общую проблему непрерывного обучения до более легкого поэтапного изучения задач. В таком случае новые задачи поступают по одной, и обучение выполняется на доступных обучающих данных.

Фактически это упрощенный вариант истинной системы непрерывного обучения, которая, скорее всего, будет встречаться на практике (De Lange et al., 2019). Например, при поэтапном обучении готовую модель обновляют, чтобы она научилась распознавать новые классы, сохраняя при этом знания о предыдущих. Схема такого подхода изображена на рис. 8.7. Говоря более формально, мы рассматриваем t -й инкрементный шаг p (где $t = 1, 2, \dots, T_{max}$), и у нас есть предыдущая модель \mathcal{M}_{t-1} и два набора данных $\{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}\}$, рандомно выбранных из распределения $\mathcal{D}^{(t)}$, которое является наблюдением (или подмножеством) полного домена \mathcal{D} . Здесь $\mathcal{X}^{(t)}$ обозначает набор образцов данных для шага t , а $\mathcal{Y}^{(t)}$ – соответствующие эталонные аннотации (т. е. одну метку для задачи классификации изображений или карту плотных меток для

задачи семантической сегментации). В рассматриваемой здесь пошаговой настройке модели под новые классы мы предполагаем, что каждый шаг соответствует отдельной обучающей задаче.

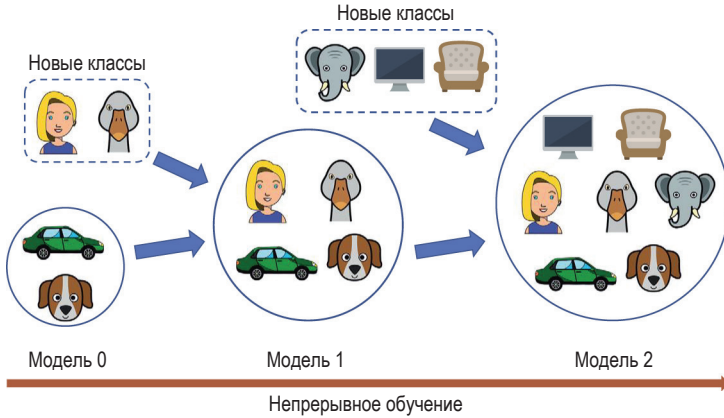


Рис. 8.7 ❖ Графическое представление схемы непрерывного обучения с приращением новых классов. Модель обновляется, чтобы научиться распознавать новые классы, не забывая ранее изученные

Чтобы имитировать ситуации реального применения и уменьшить потребность в хранении пользовательских данных или обойти ограничения конфиденциальности, большинство фреймворков не хранят никакие наборы данных $\{\mathcal{X}^{(s)}, \mathcal{Y}^{(s)}\}$ для любого шага s , предшествующего текущему шагу t . Следовательно, проблема становится еще более сложной, поскольку цель состоит в том, чтобы контролировать целевую функцию, состоящую из всех наблюдаемых задач, без доступа к предыдущим выборкам. В более формальном представлении эмпирический механизм минимизации риска преобразуется в исследование оптимальных параметров θ^* путем оптимизации:

$$\operatorname{argmin}_{\theta} \sum_{t=0}^T \mathbb{E}_{(\mathcal{X}^{(s)}, \mathcal{Y}^{(s)})} [\mathcal{L}(\mathcal{M}_t(\mathcal{X}^{(t)}; \theta), \mathcal{Y}^{(t)})] \quad (8.1)$$

с параметрами модели θ , функцией потерь \mathcal{L} , T дополнительных задач, наблюдаемых до текущего момента, и \mathcal{M}_t признаков модели на шаге t . Заметим, однако, что эта целевая функция не может быть оптимизирована напрямую, поскольку старые образцы могут вообще не присутствовать или могут быть очень ограниченными (в зависимости от сценария непрерывного обучения, см. раздел 8.3.2). Мы рассматриваем случай, когда доступны все образцы, как *совмещенное обучение* (joined learning), представляющее верхний предел качества системы непрерывного обучения (т. е. один этап обучения со всеми образцами сразу).

Кроме того, полезно получить представление о проблеме с точки зрения предельных выходных и входных распределений, то есть $P(\mathcal{Y}^{(t)})$ и $P(\mathcal{X}^{(t)})$ соответственно, для обобщенного шага t . В общем случае инкрементное обуче-

ние модели новым задачам предполагает, что $P(\mathcal{Y}^{(t+1)}) \neq P(\mathcal{Y}^{(t)})$, поскольку $P(\mathcal{X}^{(t+1)}) \neq P(\mathcal{X}^{(t)})$ и выходное пространство задачи меняется со временем, т. е. $\{\mathcal{Y}^{(t)}\} \neq \{\mathcal{Y}^{(t+1)}\}$. Продолжая нашу аналогию со сценарием UDA, мы могли бы свести пошаговое обучение новой задаче обратно к UDA, принимая $P(\mathcal{Y}^{(t+1)}) \neq P(\mathcal{Y}^{(t)})$ из-за $P(\mathcal{X}^{(t+1)}) \neq P(\mathcal{X}^{(t)})$, но $\{\mathcal{Y}^{(t)}\} = \{\mathcal{Y}^{(t+1)}\}$ с количеством инкрементальных шагов $T_{\max} = 1$ и обычно внезапным изменением в распределении данных домена, в то время как изменения при непрерывном обучении в целом более постепенны (De Lange et al., 2019; Hsu et al., 2018).

Наконец, отметим, что идеальные схемы непрерывного обучения рассматривают бесконечный и непрерывный поток обучающих данных, и на каждом этапе система получает несколько новых образцов, полученных без соблюдения условия независимости и одинакового распределения из текущего распределения $\mathcal{D}^{(t)}$, которое само может подвергаться внезапным или постепенным изменениям без уведомления. Именно на эту схему должны быть ориентированы методы, разработанные в будущем.

8.3.2. Особенности непрерывного обучения в семантической сегментации

Несмотря на то что это совсем новая область, непрерывное обучение семантической сегментации уже встречается в разных вариантах. В частности, существующие работы различаются тем, как в них рассматривают доменные распределения $\mathcal{D}^{(t)}$ и наборы данных $\{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}\}$. Наличие разных вариантов обусловлено разными целевыми применениями. Обозначим через $\mathcal{S}^{(t-1)}$ предыдущий набор меток, который расширяется набором новых классов $\mathcal{C}^{(t)}$ на шаге t , что дает новый набор меток $\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} \cup \mathcal{C}^{(t)}$. Как обычно предполагается в инкрементных методах, наборы новых меток, обнаруживаемых на каждом шаге, не пересекаются, за исключением специального класса *фоновых меток* background или *пустого класса* void, поведение и значение которых зависят от выбранного сценария. Существует множество возможных сценариев, и одно из ключевых отличий заключается в том, как рассматривается фоновый класс, что типично для многих бенчмарков семантической сегментации. Существующие подходы относятся к одному из четырех основных сценариев:

1. **Последовательный маскированный.** Этот вариант отражает простейшую идею непрерывной семантической сегментации, т. е. каждый шаг обучения содержит уникальный набор изображений, пиксели которых принадлежат либо к новым классам, либо к пустому классу, который не предсказывается моделью и маскируется как в результатах, так и из процедуры обучения. Данный механизм описан в (Tasar et al., 2019 г.; Klingner et al., 2020).
2. **Последовательный.** Этот вариант был предложен Микьели и Зануттигом (Michieli, Zanuttigh, 2019, 2021). Каждый шаг обучения содержит уникальный набор изображений, пиксели которых принадлежат классам, наблюдаемым либо на текущем, либо на предыдущих этапах

обучения. На каждом шаге присутствуют метки для пикселей как новых классов, так и старых; однако конкретное появление определенного старого класса сильно коррелирует с набором добавляемых классов. Например, если набор всех старых классов $\mathcal{S}^{(t-1)} = \{\text{стул, самолет}\}$, а набор добавляемых классов есть $\mathcal{C}^{(t)} = \{\text{обеденный стол}\}$, то разумно ожидать, что $\{\mathcal{X}^{(t)}, \mathcal{Y}^{(t)}\}$ содержит некоторые изображения, относящиеся к классу *стульев*, который обычно встречается вместе с *обеденным столом*, в то время как изображения *самолета* крайне маловероятны.

3. **Непересекающийся.** Этот вариант был предложен (Cermelli et al., 2020; Michieli, Zanuttigh, 2021). На каждом этапе обучения уникальный набор изображений такой же, как в последовательном варианте. Отличие заключается в наборе меток. На каждом шаге присутствуют только метки для пикселей новых классов, в то время как старые помечаются на картах сегментации как фон (это приводит к изменению распределения фонового класса на каждом шаге).
4. **Перекрывающийся.** Этот вариант заимствован из работы Шмелькова и др. (Shmelkov et al., 2017), посвященной обнаружению объектов, и рассмотрен в работах (Cermelli et al., 2020; Douillard et al., 2021; Michieli, Zanuttigh, 2021) применительно к семантической сегментации. В этой настройке каждый шаг обучения содержит все изображения, которые имеют хотя бы один пиксель нового набора классов, причем только классы этого набора аннотированы, а остальные считаются фоном. В отличие от других вариантов, в этом сценарии изображения могут содержать пиксели классов, которые будут изучены в будущем, но помечены как фон на текущем шаге; по этой причине, как и в предыдущем случае, фоновый класс меняет распределение на каждом шаге итерации.

Несколько примеров различных аннотаций семантической карты приведены на рис. 8.8. Хотя они являются подкатегориями одной и той же задачи, они приводят к существенно разным ситуациям, требующим различных стратегий для решения.

Этот многовариантный сценарий становится еще более гибким, если учесть, что существует множество различных способов формирования наборов $\mathcal{C}^{(t)}$ незнакомых классов и выбора их кардинальности $|\mathcal{C}^{(t)}|$, что заставило исследователей провести множество совершенно разных экспериментов. Например, давайте рассмотрим один из наиболее широко используемых тестов для семантической сегментации – набор данных Pascal VOC2012 (Everingham et al., 2010), который состоит из 21 семантического класса (включая фон). Что касается первого аспекта, одна из возможностей состоит в том, чтобы отсортировать классы, используя заранее определенный порядок, предоставленный набором данных (например, алфавитный порядок для VOC2012), как это сделано в (Shmelkov et al., 2017; Michieli, Zanuttigh, 2019, 2021; Cermelli et al., 2020; Douillard et al., 2021; Michieli, Zanuttigh, 2021). Другой возможностью является сортировка классов на основе их встречаемости в наборе данных (Michieli, Zanuttigh, 2021), чтобы отразить идею о том, что в реальных приложениях было бы разумно начать с общих классов, а затем ввести более редкие. Что касается второго аспекта, то можно последовательно добавлять

один класс, группу классов или несколько классов (Michieli, Zanuttigh, 2019, 2021; Cermelli et al., 2020; Douillard et al., 2021; Michieli, Zanuttigh, 2021). Все эти возможности образуют весьма пеструю картину, к исследованию которой приступили лишь недавно, поэтому многие направления остаются до сих пор неизученными.

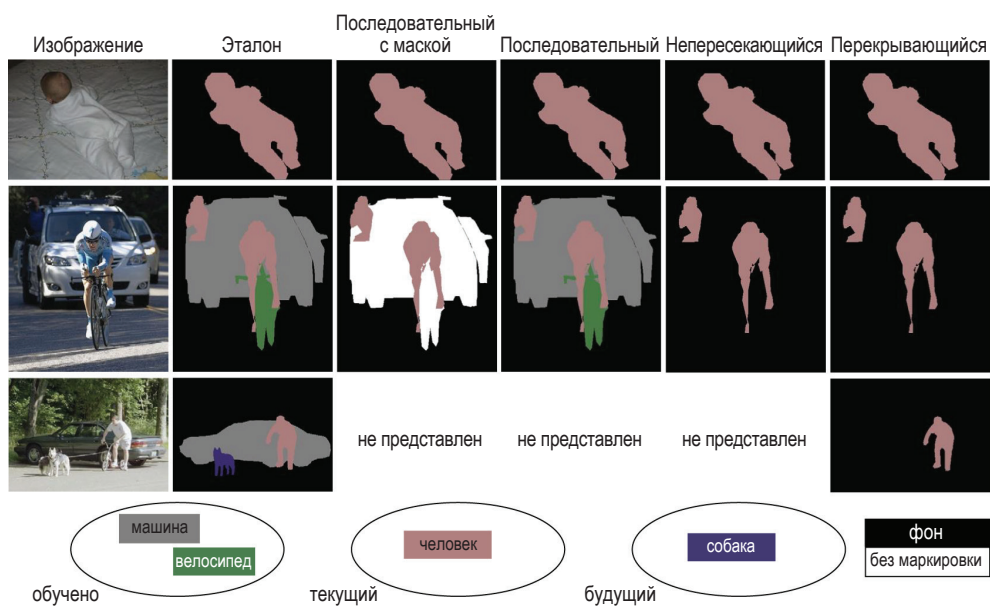


Рис. 8.8 ❖ Обзор различных сценариев непрерывного обучения с добавлением класса при семантической сегментации. Черный цвет представляет фоновый класс background, а белый – пустой/непомеченный класс void/unlabeled

8.3.3. Методы поэтапного обучения

В этом разделе мы рассмотрим основные методы решения инкрементной задачи семантической сегментации, сгруппированные по используемой технике. Мы также предлагаем заинтересованным читателям самостоятельно ознакомиться с некоторыми соответствующими работами по инкрементной классификации изображений, поскольку эта родственная область является более изученной и зрелой в отношении семантической сегментации.

8.3.3.1. Дистилляция знаний

Мы начнем с метода, который используют наиболее часто благодаря его простоте и эффективности, – это *дистилляция знаний* (knowledge distillation). Этот метод изначально был предложен (Bucilua et al., 2006) и (Hinton et al., 2015), чтобы сохранить выходные данные сложного ансамбля сетей при переходе на более простую сеть для более эффективного ее развертывания. Впоследствии идея была адаптирована для сохранения неизменным

отклика сети на старые задачи при обновлении ее новыми обучающими выборками, обычно связанными с новыми задачами. Как правило, этого добиваются путем применения ограничения (например, функции потерь), чтобы имитировать ответы предыдущей модели в ее текущей версии. Основным эффектом ограничения заключается в его действии как мощного фактора регуляризации в процессе изучения текущих классов, что часто приводит к лучшим результатам как на предыдущих, так и на текущих классах (за счет сохранения способности распознавать первый набор и избегать переоценки последнего набора).

Дистилляция знаний изучалась в различных условиях, и в некотором роде она является обязательным компонентом успешных алгоритмов инкрементного обучения новым задачам. Многие алгоритмы используют дистилляцию знаний в разных вариантах задачи с разреженной разметкой: например, в работе (Shmelkov et al., 2017) предлагают сквозную схему обучения, в которой представление и классификатор обучаются совместно без сохранения каких-либо исходных обучающих образцов. Ли и Хойем (Li, Hoiem, 2017) извлекают предыдущие знания непосредственно из последней обученной модели. Дхар и др. (Dhar et al., 2019) вводят потерю дистилляции внимания в качестве штрафа за сохранение информации для карт внимания классификаторов. Чжоу и др. (Zhou et al., 2019) извлекают из всех предыдущих снимков модели знания, из которых составляется сокращенная версия.

Было обнаружено, что эти методы чрезвычайно эффективны и надежны даже при выполнении сложных задач. Оздемир и Гоксель (Ozdemir and Goksel, 2019) расширяют модель классификации изображений Ли и Хойема (Li, Hoiem, 2017) до сегментации, просто выражая потери при дистилляции знаний как перекрестную энтропию между вероятностями вывода предыдущей и текущей моделей. Авторы также разрабатывают стратегию выбора соответствующих образцов старых данных для повторного использования, что улучшает качество модели, но нарушает предположение многих сценариев об отказе от хранения предыдущих данных. Тасар и др. (Tasar et al., 2019) применяют дистилляцию знаний посредством перекрестной энтропии между выходными вероятностями предыдущей и текущей моделей для каждого класса, поскольку модель прогнозирует карты бинарной сегментации для каждого класса отдельно. Микьели и Зануттиг (Michieli, Zanuttigh, 2019) оценивают стандартный бенчмарк семантической сегментации и предлагают применять дистилляцию знаний не только на уровне вывода, но и на промежуточном пространстве признаков, чтобы сохранить геометрические отношения извлеченных признаков. Работа расширена в (Michieli, Zanuttigh, 2021), где представлены и сравниваются многие методы дистилляции знаний. В частности, дистилляция на выходном слое обогащена температурным масштабированием (т. е. масштабированием вероятностей softmax с помощью так называемого *температурного коэффициента*), чтобы учесть также неопределенность оценок предыдущих моделей. В работе (Tung, Mori, 2019) дистилляция на промежуточном уровне признаков расширена до нескольких этапов декодирования, а также предлагается схема на основе дистилляции знаний с сохранением подобия. Чермелли и др. (Cermelli et al., 2020) предлагают рассмотреть потери при дистилляции на уровне вывода,

учитывающие тот факт, что предыдущая модель могла уже быть знакома с предыдущими классами, помеченными как фон (т. е. упомянутый ранее вариант с перекрытием классов). Клингнер и др. (Klingner et al., 2020) предлагают учитывать маскированные и взвешенные потери при дистилляции на уровне выхода, чтобы повысить точность небольших или недостаточно представленных классов в наборе данных. Наконец, Дуйяр и др. (Douillard et al., 2021) применяют дистилляцию, чтобы сохранить статистику на разных уровнях признаков и в разных масштабах между старой и текущей моделями.

8.3.3.2. Замораживание параметров

К главным достижениям ранних исследований сетевых моделей относится обнаружение одной из основных стратегий решения проблемы катастрофического забывания – *замораживание* части весов сети (Rebuffi et al., 2017). Этот метод применялся во многих современных подходах в качестве попытки регуляризации для предотвращения деградации знаний, вызванной будущими задачами. Например, Шмельков и др. (2017) проводят эксперимент по замораживанию либо всех слоев (кроме последнего), либо их части. Мандзюк и Шастри (Mandziuk, Shastri, 2002) пытаются найти и заморозить компактное подмножество признаков (узлов) в скрытых слоях, имеющих решающее значение для текущей задачи, тем самым предотвращая забывание в будущем. Аналогичным образом Киркпатрик и др. (Kirkpatrick et al., 2017) запоминают старые задачи, замедляя процесс обучения на соответствующих весовых коэффициентах для этих задач. Юнг и др. (Jung et al., 2016) пытаются сохранить качество модели на старых задачах, замораживая последний слой и препятствуя изменению общих весов в слоях извлечения признаков.

Замораживание параметров как способ предотвращения забывания было также предложено в задаче плотной разметки. Микьели и Зануттиг (2019) предлагают заморозить все уровни кодировщика, чтобы сохранить неизменными возможности извлечения признаков и обучать только параметры декодирования. Эти же авторы используют идею замораживания только первых нескольких слоев кодировщика, чтобы сохранить наиболее независимую от задачи часть экстрактора признаков. Однако вопрос о том, какие слои следует заморозить, остается открытым, и существует внутренний компромисс между возможностью эффективного изучения новых задач и сохранением полученных знаний. Первая попытка автоматического выбора слоев для замораживания на основании поиска самых пластичных слоев сети была недавно предложена в работе (Nguyen et al., 2020).

8.3.3.3. Геометрическая регуляризация на уровне признаков

Анализ организации скрытого пространства приобретает решающее значение для понимания и улучшения глубоких нейронных сетей (Bengio et al., 2013; Girshick et al., 2014; Xian et al., 2016; Peng et al., 2019). В последнее время определенное внимание уделяется скрытой регуляризации при непрерывной классификации изображений (Achille et al., 2018; Javed, White, 2019) и адаптации домена без учителя (Toldo et al., 2021; Barbato et al., 2021).

Ключевая идея этих подходов состоит в том, чтобы по-разному разделить промежуточное пространство признаков, разнести признаки разных классов. При непрерывном обучении это может уменьшить перекрытие, когда в модель вводятся будущие классы.

Единственная работа, использующая эту идею в сложных задачах, – это (Michieli, Zanuttigh, 2021), где скрытое пространство ограничивают, чтобы уменьшить забывание и улучшить распознавание новых классов. Эта схема зависит от трех основных компонентов: во-первых, совпадение прототипов обеспечивает целостность скрытого пространства для старых классов, заставляя кодировщик создавать аналогичные скрытые представления для ранее просмотренных классов на последующих этапах; во-вторых, прореживание признаков позволяет освободить место в скрытом пространстве для размещения новых классов; в-третьих, для группировки признаков в соответствии с их семантикой используется контрастное обучение, при этом разделяют признаки разных классов.

8.3.3.4. Новые направления

В публикациях по непрерывному обучению были предложены и другие новые идеи как для плотной, так и для разреженной маркировки. Далее мы представляем лишь некоторые из наиболее перспективных направлений исследований.

Инициализация весов использовалась в (Cermelli et al., 2020), чтобы справиться с нетипичным поведением фоновых классов в сценариях с непересекающимися и перекрывающимися классами. Авторы инициализируют параметры классификатора для новых классов таким образом, что вероятность фона равномерно распределяется между новыми классами, предотвращая смещение модели в сторону фоновых классов при работе с незнакомыми классами.

Методы **генеративного воспроизведения** (generative replay) обучают генеративные модели текущему распределению данных; после этого можно генерировать имитации данных из прошлого опыта при изучении новых данных. Изучая реальные новые данные, смешанные с искусственно сгенерированными прошлыми данными, модели пытаются сохранить прошлые знания при изучении новой задачи. Генеративная модель обычно представляет собой GAN (Goodfellow et al., 2014), как в (Wu et al., 2018) и (Shin et al., 2017), или автокодировщик, как у (Draelos et al., 2017) и (Kamra et al., 2017). Обратите внимание, что для сгенерированных данных доступны только слабые метки классов, и для сегментации необходимо вычислять некоторые псевдометки.

Интернет-обучаемые модели (Hou et al., 2018; Modolo, Ferrari, 2017), т. е. модели, которые учатся на образцах, полученных с помощью веб-поиска, могут быть чрезвычайно мощным инструментом для получения достоверных старых образцов с использованием в качестве запросов имен меток старых классов, которые нужно сохранить. В этом случае также доступны только слабые метки классов, и необходимо добавить в модель какую-то схему псевдомаркировки.

8.4. ЗАКЛЮЧЕНИЕ

Семантическая сегментация изображений является активной областью исследований, направленных на достижение детального и точного понимания сцены. Будучи задачей плотной разметки, она обладает дополнительной сложностью по сравнению с классическими задачами классификации изображений. Для решения этой задачи было предложено множество моделей глубокого обучения, однако их архитектуры требуют больших аннотированных обучающих наборов данных и демонстрируют плохие возможности адаптации к незнакомым доменам или задачам. В последние годы исследователи ведут активную деятельность в области адаптации доменов и непрерывного обучения, направленную на устранение упомянутых ограничений. В этой главе мы рассмотрели адаптацию домена без учителя (т. е. когда для обучения не используются размеченные данные целевого домена) применительно к задаче плотной семантической сегментации. Затем мы показали, что для решения этой задачи можно применить непрерывное обучение, организованное несколькими разными способами.

Алгоритмы, разработанные к настоящему времени, способны значительно уменьшить деградацию модели, хотя им все еще необходимо достичь большей зрелости, прежде чем их можно будет применять в ответственных реальных сценариях (например, в автономном вождении). На самом деле осталось много нерешенных проблем в плане как адаптации сложных архитектур глубокого обучения к различным задачам и областям, так и их способности изучать новые концепции, не забывая ранее полученные знания.

Кроме того, в реальных сценариях использования моделей возникает много сопутствующих побочных проблем. К перспективным направлениям можно отнести, например, обобщение из нескольких распределений данных, UDA с открытым набором классов (т. е. распознавание классов в целевой области, никогда не встречавшихся в исходной) и непрерывный UDA (т. е. непрерывный процесс адаптации к незнакомым доменам и задачам).

БЛАГОДАРНОСТИ

Наша работа была частично поддержана итальянским министерством образования (MIUR) в рамках инициативы «Направления передового опыта» (Закон 232/2016).

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

Achille A., Eccles T., Matthey L., Burgess C., Watters N., Lerchner A., Higgins I., 2018. Life-long disentangled representation learning with cross-domain latent homologies. In: Neural Information Processing Systems (NeurIPS).

- Barbato F., Toldo M., Michieli U., Zanuttigh P.*, 2021. Latent space regularization for unsupervised domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
- Bengio Y., Courville A., Vincent P.*, 2013. Representation learning: a review and new perspectives. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). IEEE, pp. 1798–1828.
- Biasetton M., Michieli U., Agresti G., Zanuttigh P.*, 2019. Unsupervised domain adaptation for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
- Bucher M., Vu T. H., Cord M., Pérez P.*, 2020. Buda: boundless unsupervised domain adaptation in semantic segmentation. arXiv preprint. arXiv:2004.01130.
- Bucilua C., Caruana R., Niculescu-Mizil A.*, 2006. Model compression. In: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 535–541.
- Busto P. P., Gall J.*, 2017. Open set domain adaptation. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 754–763.
- Castro F. M., Marín-Jiménez M. J., Guil N., Schmid C., Alahari K.*, 2018. End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 233–248.
- Cermelli F., Mancini M., Bulò S. R., Ricci E., Caputo B.*, 2020. Modeling the background for incremental learning in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Chang W., Wang H., Peng W., Chiu W.*, 2019. All about structure: adapting structural information across domains for boosting semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1900–1909.
- Chen Y., Li W., Chen X., Van Gool L.*, 2018. Learning semantic segmentation from synthetic data: a geometrically guided input-output adaptation approach. arXiv preprint. arXiv:1812.05040.
- Chen M., Xue H., Cai D.*, 2019a. Domain adaptation for semantic segmentation with maximum squares loss. In: Proceedings of the International Conference on Computer Vision (ICCV).
- Chen Y., Li W., Chen X., Gool L. V.*, 2019b. Learning semantic segmentation from synthetic data: a geometrically guided input-output adaptation approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1841–1850.
- Chen Y. C., Lin Y. Y., Yang M. H., Huang J. B.*, 2019c. Crdoco: pixel-level domain transfer with cross-domain consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Chen Y. H., Chen W. Y., Chen Y. T., Tsai B. C., Frank Wang Y. C., Sun M.*, 2017. No more discrimination: cross city adaptation of road scene segmenters. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1992–2001.
- Choi J., Kim T., Kim C.*, 2019. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 6830–6840.

- De Lange M., Aljundi R., Masana M., Parisot S., Jia X., Leonardis A., Slabaugh G., Tuytelaars T., 2019. Continual learning: a comparative study on how to defy forgetting in classification tasks. arXiv preprint. arXiv:1909.08383.
- Deng J., Dong W., Socher R., Li L., Li K., Li F., 2009. Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255.
- Dhar P., Singh R. V., Peng K. C., Wu Z., Chellappa R., 2019. Learning without memorizing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5138–5146.
- Ditzler G., Roveri M., Alippi C., Polikar R., 2015. Learning in nonstationary environments: a survey. IEEE Computational Intelligence Magazine 10, 12–25.
- Douillard A., Chen Y., Dapogny A., Cord M., 2021. Plop: learning without forgetting for continual semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Draeos T. J., Miner N. E., Lamb C. C., Cox J. A., Vineyard C. M., Carlson K. D., Severa W. M., James C. D., Aimone J. B., 2017. Neurogenesis deep learning: extending deep networks to accommodate new classes. In: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 526–533.
- Du L., Tan J., Yang H., Feng J., Xue X., Zheng Q., Ye X., Zhang X., 2019. SSF-DAN: separated semantic feature based domain adaptation network for semantic segmentation. In: Proceedings of the International Conference on Computer Vision (ICCV).
- Dundar A., Liu M., Wang T., Zedlewski J., Kautz J., 2018. Domain stylization: a strong, simple baseline for synthetic to real image domain adaptation. arXiv preprint. arXiv:1807.09384.
- Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A., 2010. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88, 303–338.
- French R. M., 1999. Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences 3, 128–135.
- Ganin Y., Lempitsky V., 2015. Unsupervised domain adaptation by backpropagation. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1180–1189.
- Ganin Y., Ustinova E., Ajakan H., Germain P., Larochelle H., Laviolette F., Marchand M., Lempitsky V., 2016. Domain-adversarial training of neural networks. Journal of Machine Learning Research 17, 2096–2130.
- Girshick R., Donahue J., Darrell T., Malik J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587.
- Gong R., Li W., Chen Y., Gool L. V., 2019. DLOW: domain flow for adaptation and generalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2477–2486.
- Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014. Generative adversarial nets. In: Neural Information Processing Systems (NeurIPS), pp. 2672–2680.
- Grandvalet Y., Bengio Y., 2005. Semi-supervised learning by entropy minimization. In: Actes de CAP 05, Conférence francophone sur l'apprentissage automatique, pp. 281–296.

- Grossberg S., 2013. Adaptive resonance theory: how a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks* 37, 1–47.
- Hinton G., Vinyals O., Dean J., 2015. Distilling the knowledge in a neural network. arXiv preprint. arXiv:1503.02531.
- Hoffman J., Tzeng E., Park T., Zhu J. Y., Isola P., Saenko K., Efros A., Darrell T., 2018. Cycada: cycle-consistent adversarial domain adaptation. In: *Proceedings of the International Conference on Machine Learning (ICML)*.
- Hoffman J., Wang D., Yu F., Darrell T., 2016. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint. arXiv:1612.02649.
- Hou Q., Cheng M. M., Liu J., Torr P. H., 2018. Webseg: learning semantic segmentation from web searches. arXiv preprint. arXiv:1803.09859.
- Hsu Y. C., Liu Y. C., Ramasamy A., Kira Z., 2018. Re-evaluating continual learning scenarios: a categorization and case for strong baselines. arXiv preprint. arXiv:1810.12488.
- Javed K., White M., 2019. Meta-learning representations for continual learning. In: *Neural Information Processing Systems (NeurIPS)*.
- Jung H., Ju J., Jung M., Kim J., 2016. Less-forgetting learning in deep neural networks. arXiv preprint. arXiv: 1607.00122.
- Kamra N., Gupta U., Liu Y., 2017. Deep generative dual memory network for continual learning. arXiv preprint. arXiv:1710.10368.
- Kemker R., McClure M., Abitino A., Hayes T. L., Kanan C., 2018. Measuring catastrophic forgetting in neural networks. In: *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Kirkpatrick J., Pascanu R., Rabinowitz N., Veness J., Desjardins G., Rusu A. A., Milan K., Quan J., Ramalho T., Grabska-Barwinska A., et al., 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)* 114, 3521–3526.
- Klingner M., Bär A., Donn P., Fingscheidt T., 2020. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*.
- Kundu J. N., Venkatesh R. M., Venkat N., Revanur A., Babu R. V., 2020. Class-incremental domain adaptation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Lee K., Ros G., Li J., Gaidon A., 2019a. SPIGAN: privileged adversarial learning from simulation. In: *International Conference on Learning Representations (ICLR)*.
- Lee S., Kim D., Kim N., Jeong S. G., 2019b. Drop to adapt: learning discriminative features for unsupervised domain adaptation. In: *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 91–100.
- Lesort T., Lomonaco V., Stoian A., Maltoni D., Filliat D., Díaz-Rodríguez N., 2020. Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. *Information Fusion* 58, 52–68.
- Li P., Liang X., Jia D., Xing E. P., 2018. Semantic-aware grad-gan for virtual-to-real urban scene adaption. In: *Proceedings of British Machine Vision Conference (BMVC)*.
- Li Y., Yuan L., Vasconcelos N., 2019. Bidirectional learning for domain adaptation of semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Li Z., Hoiem D., 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 2935–2947.
- Long M., Cao Y., Wang J., Jordan M., 2015. Learning transferable features with deep adaptation networks. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 97–105.
- Luo Y., Zheng L., Guan T., Yu J., Yang Y., 2019. Taking a closer look at domain shift: category-level adversaries for semantics consistent domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Mańdziuk J., Shastri L., 2002. Incremental class learning approach and its application to handwritten digit recognition. *Information Sciences* 141, 193–217.
- McClelland J. L., McNaughton B. L., O'Reilly R. C., 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review* 102, 419.
- McCloskey M., Cohen N. J., 1989. Catastrophic interference in connectionist networks: the sequential learning problem. In: *Psychology of Learning and Motivation*, vol. 24. Elsevier, pp. 109–165.
- Michieli U., Basetton M., Agresti G., Zanuttigh P., 2020. Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation. *IEEE Transaction on Intelligent Vehicles*.
- Michieli U., Zanuttigh P., 2019. Incremental learning techniques for semantic segmentation. In: *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*.
- Michieli U., Zanuttigh P., 2021a. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Michieli U., Zanuttigh P., 2021b. Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding* 205, 103167.
- Modolo D., Ferrari V., 2017. Learning semantic part-based models from Google images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, 1502–1509.
- Murez Z., Kolouri S., Kriegman D. J., Ramamoorthi R., Kim K., 2018. Image to image translation for domain adaptation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nguyen G., Chen S., Do T., Jun T. J., Choi H. J., Kim D., 2020. Dissecting catastrophic forgetting in continual learning by deep visualization. *arXiv preprint. arXiv:2001.01578*.
- Ozdemir F., Goksel O., 2019. Extending pretrained segmentation networks with additional anatomical structures. *International Journal of Computer Assisted Radiology and Surgery* 14, 1187–1195.
- Parisi G. I., Kemker R., Part J. L., Kanan C., Wermter S., 2019. Continual lifelong learning with neural networks: a review. *Neural Networks*.
- Peng X., Huang Z., Sun X., Saenko K., 2019. Domain agnostic learning with disentangled representations. In: *Proceedings of the International Conference on Machine Learning (ICML)*, PMLR, pp. 5102–5112.

- Pizzati F., Charette R. D., Zaccaria M., Cerri P., 2020. Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV), pp. 2990–2998.
- Qin C., Wang L., Zhang Y., Fu Y., 2019. Generatively inferential co-training for unsupervised domain adaptation. In: Proceedings of the International Conference on Computer Vision Workshops (ICCVW), pp. 1055–1064.
- Rebuffi S. A., Kolesnikov A., Sperl G., Lampert C. H., 2017. Icarl: incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2001–2010.
- Saito K., Kim D., Sclaroff S., Saenko K., 2020. Universal domain adaptation through self-supervision. In: Neural Information Processing Systems (NeurIPS).
- Saito K., Ushiku Y., Harada T., Saenko K., 2018a. Adversarial dropout regularization. In: International Conference on Learning Representations (ICLR).
- Saito K., Watanabe K., Ushiku Y., Harada T., 2018b. Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3723–3732.
- Sankaranarayanan S., Balaji Y., Jain A., Nam Lim S., Chellappa R., 2018. Learning from synthetic data: addressing domain shift for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3752–3761.
- Shin H., Lee J. K., Kim J., Kim J., 2017. Continual learning with deep generative replay. In: Neural Information Processing Systems (NeurIPS), pp. 2990–2999.
- Shmelkov K., Schmid C., Alahari K., 2017. Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 3400–3409.
- Spadotto T., Toldo M., Michieli U., Zanuttigh P., 2020. Unsupervised domain adaptation with multiple domain discriminators and adaptive self-training. In: Proceedings of the IEEE International Conference on Pattern Recognition (ICPR).
- Tasar O., Tarabalka Y., Alliez P., 2019. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12, 3524–3537.
- Thrun S., Pratt L., 2012. *Learning to Learn*. Springer Science & Business Media.
- Toldo M., Michieli U., Agresti G., Zanuttigh P., 2020. Unsupervised domain adaptation for mobile semantic segmentation based on cycle consistency and feature alignment. *Image and Vision Computing*.
- Toldo M., Michieli U., Zanuttigh P., 2021. Unsupervised domain adaptation in semantic segmentation via orthogonal and clustered embeddings. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV).
- Tsai Y. H., Hung W. C., Schuster S., Sohn K., Yang M. H., Chandraker M., 2018. Learning to adapt structured output space for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7472–7481.
- Tsai Y. H., Sohn K., Schuster S., Chandraker M., 2019. Domain adaptation for structured output via discriminative patch representations. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1456–1465.

- Tung F., Mori G.*, 2019. Similarity-preserving knowledge distillation. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1365–1374.
- Vu T., Jain H., Bucher M., Cord M., Pérez P.*, 2019a. DADA: depth-aware domain adaptation in semantic segmentation. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 7363–7372.
- Vu T. H., Jain H., Bucher M., Cord M., Pérez P.*, 2019b. Advent: adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2517–2526.
- Wang M., Deng W.*, 2018. Deep visual domain adaptation: a survey. *Neurocomputing* 312, 135–153.
- Watanabe K., Saito K., Ushiku Y., Harada T.*, 2018. Multichannel semantic segmentation with unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV).
- Wu Y., Chen Y., Wang L., Ye Y., Liu Z., Guo Y., Zhang Z., Fu Y.*, 2018. Incremental classifier learning with generative adversarial networks. *arXiv preprint. arXiv:1802.00853*.
- Wu Z., Wang X., Gonzalez J., Goldstein T., Davis L.*, 2019. ACE: adapting to changing environments for semantic segmentation. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 2121–2130.
- Xian Y., Akata Z., Sharma G., Nguyen Q., Hein M., Schiele B.*, 2016. Latent embeddings for zero-shot classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 69–77.
- Yang J., An W., Wang S., Zhu X., Yan C., Huang J.*, 2020a. Label-driven reconstruction for domain adaptation in semantic segmentation. *arXiv preprint. arXiv:2003.04614*.
- Yang Y., Lao D., Sundaramoorthi G., Soatto S.*, 2020b. Phase consistent ecological domain adaptation. *arXiv preprint. arXiv:2004.04923*.
- Yang Y., Soatto S.*, 2020. FDA: Fourier domain adaptation for semantic segmentation. *arXiv preprint. arXiv:2004.05498*.
- Zhang Y., Qiu Z., Yao T., Liu D., Mei T.*, 2018. Fully convolutional adaptation networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6810–6818.
- Zhou P., Mai L., Zhang J., Xu N., Wu Z., Davis L. S.*, 2019. M2kd: multi-model and multi-level knowledge distillation for incremental learning. *arXiv preprint. arXiv:1904.01769*.
- Zhou Q., Feng Z., Cheng G., Tan X., Shi J., Ma L.*, 2020. Uncertainty-aware consistency regularization for crossdomain semantic segmentation. *arXiv preprint. arXiv:2004.08878*.
- Zhu J., Park T., Isola P., Efros A. A.*, 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the International Conference on Computer Vision (ICCV).
- Zhu X., Zhou H., Yang C., Shi J., Lin D.*, 2018. Penalizing top performers: conservative loss for semantic segmentation adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 568–583.

- Zhuo J., Wang S., Cui S., Huang Q.*, 2019. Unsupervised open domain recognition by semantic discrepancy minimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 750–759.
- Zou Y., Yu Z., Liu X., Kumar B. V., Wang J.*, 2019. Confidence regularized self-training. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 5982–5991.
- Zou Y., Yu Z., Vijaya Kumar B., Wang J.*, 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 289–305.

ОБ АВТОРАХ ГЛАВЫ

Умберто Микьели получил степень магистра в области телекоммуникаций в Университете Падуи в 2018 г. На момент написания данной работы он был студентом выпускного курса того же университета. В 2018 г. провел 6 месяцев в качестве приглашенного исследователя в Техническом университете Дрездена. В 2020 г. он в течение 8 месяцев проходил стажировку в качестве инженера-исследователя в Samsung Research UK. Его исследования сосредоточены на методах переноса обучения для семантической сегментации, в частности на адаптации домена и непрерывном обучении.

Марко Тольдо получил степень магистра в области информационно-коммуникационных технологий для интернета и мультимедиа в 2019 г. в Университете Падуи. В настоящее время работает над докторской диссертацией на кафедре информационной инженерии того же университета. В 2021 г. в течение 7 месяцев проходил стажировку в качестве инженера-исследователя в Samsung Research UK. Его исследовательские интересы включают адаптацию домена и непрерывное обучение применительно к компьютерному зрению.

Пьетро Зануттиг получил степень магистра в области вычислительной техники и докторскую степень в Университете Падуи в 2003 и 2007 гг. соответственно. В настоящее время является доцентом кафедры информационных технологий того же университета. Его исследовательские интересы включают семантическое понимание изображений и 3D-данных, адаптацию домена и инкрементное обучение для обработки визуальных данных, обработку 3D-данных с особым акцентом на датчики ToF, слияние данных с нескольких датчиков и распознавание жестов рук.

Глава 9

Визуальное отслеживание движущихся объектов

Автор главы:

Майкл Фелсберг, Лаборатория компьютерного зрения, факультет электроники, Линчепингский университет, Линчепинг, Швеция; Инженерная школа, Университет Квазулу-Наталь, Дурбан, Южная Африка

Краткое содержание главы:

- визуальное отслеживание объектов;
- дискриминативные подходы к отслеживанию;
- корреляционные фильтры;
- глубокие признаки;
- глубокое обучение в отслеживании;
- сегментация видеообъектов;
- дискриминативная сегментация.

9.1. ВВЕДЕНИЕ

Отслеживание (tracking) – очень неоднозначный термин, и даже для визуального отслеживания существует множество различных интерпретаций и предположений по умолчанию. Эта глава начинается с тщательного и точного определения рассматриваемой задачи отслеживания, чтобы избежать недопонимания и сделать все предположения явными.

9.1.1. Определение задачи отслеживания

В этой главе мы рассмотрим задачу *общего визуального отслеживания* в том смысле, как это определено в формулировке *визуального отслеживания объектов* (visual object tracking, VOT) 2013–2020 гг. (Kristan et al., 2013, 2015a,b, 2016, 2017, 2019,b, 2020), что также совпадает с определением *бенчмарка последовательного отслеживания* (online tracking benchmark, OTB) (Wu et al., 2013). Отслеживание выполняется в домене изображения, и никакие пред-

варительные знания о классах объектов недоступны; «общий» означает не-зависимый от класса.

Задача формулируется в плоскости 2D-изображения, в отличие от подходов с 3D-трекингом (Garon, Lalonde, 2017). Кроме того, рассматривается только один объект (цель), в отличие от случая отслеживания нескольких объектов (Dendorfer et al., 2020), который требует связывания целей через последовательность изображений. Однако отслеживание одного объекта не означает, что в последовательностях кадров нет других движущихся объектов, так называемых *дистракторов* (distractor). Задача слежения требует, чтобы цель не смешивалась ни с одним из дистракторов, даже если они частично перекрывают цель. Предполагается, что цель хотя бы частично видна на протяжении всей последовательности, так что повторное обнаружение не требуется. Последовательность может исходить от движущейся камеры, что исключает простое моделирование фона для обнаружения цели (Stauffer, Grimson, 2000).

Цель отслеживания определяется одиночной аннотацией в первом кадре последовательности изображений в виде ограничивающей рамки (или маски сегментации, раздел 9.5). Задача состоит в том, чтобы предсказать ограничивающую рамку (или маску сегментации), содержащую один и тот же объект во всех последующих кадрах. Ограничивающие рамки могут быть определены в различных системах координат; конкретный формат не имеет значения, если он не меняется в процессе отслеживания. В стандартной задаче *визуального отслеживания объекта* (visual object tracking, VOT) используются левый верхний угол изображения (x, y) , ширина w и высота h (начало координат в левом верхнем углу). Таким образом, мы получаем следующее формальное определение задачи:

- *ввод*: видеопоток (последовательность изображений) и одна аннотированная ограничивающая рамка (x_0, y_0, w_0, h_0) для начального кадра последовательности;
- *вывод*: предсказание ограничивающих рамок (x_k, y_k, w_k, h_k) для всех кадров $k > 0$;
- *условие*: предсказание ограничивающей рамки для кадра K может использовать только данные из кадров $k \leq K$.

Это определение задачи отслеживания фактически сводится к задаче *однократного обучения* (one-shot learning), поскольку для данной входной последовательности предоставляется единственная аннотированная обучающая выборка. Целью этой задачи является обучение детектора, который предсказывает наилучшую ограничивающую рамку в последующем кадре, что также называется *отслеживанием через обнаружение* (tracking by detection). Как правило, наиболее подходящая ограничивающая рамка классифицируется как цель, а все остальные – как фон. Однако также можно использовать методы регрессии для определения (уточнения) параметров ограничивающей рамки.

9.1.2. Затруднения при отслеживании

Основная трудность в отслеживании через обнаружение заключается в том, что внешний вид отслеживаемого объекта может измениться на протяжении последовательности. В частности, объект может:

- вращаться в плоскости изображения;
- изменить масштаб, смещаясь по глубине;
- изменить соотношение сторон путем поворота вне плоскости / изменения точки обзора;
- изменить форму;
- страдать от размытия движения;
- претерпевать изменения освещения;
- быть частично закрытым.

Эти изменения могут привести к сбою детектора и потере цели, особенно при наличии засоренного фона и отвлекающих факторов. Поэтому современные методы отслеживания обычно адаптируют модель детектора, используя ее собственные прогнозы из предыдущих кадров (рис. 9.1).

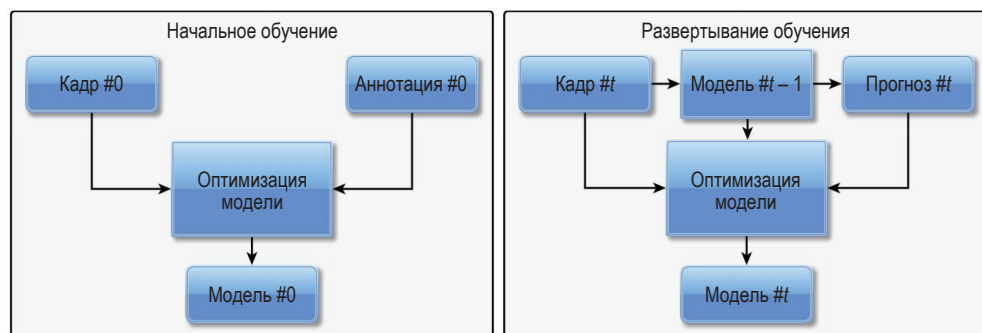


Рис. 9.1 ❖ Блок-схема адаптивной отслеживающей модели. После начального обучения (слева) модель адаптируется, используя собственные прогнозы в качестве обучающих выборок (справа)

Этот адаптивный процесс определяет процедуру *развертывания модели* (model bootstrapping), то есть рекурсивное самосовершенствование, которое происходит без внешнего ввода, кроме единственной начальной аннотации. Одна из ключевых проблем этого процесса состоит в том, чтобы найти баланс между сохранением предыдущей модели и ее обновлением (гибкость модели). Если модель слишком жесткая, она потеряет цель при изменении внешнего вида, если модель слишком гибкая, она будет дрейфовать от цели (Matthews et al., 2004; Wang et al., 2016).

9.1.3. Обоснование методики

Проблема дрейфа модели в значительной степени уменьшается, если известен класс интересующего объекта. Однако, как указано в разделе 9.1.1, мы стремимся к универсальным, т. е. не зависящим от класса, моделям. Основным обоснованием для этого выбора является допущение об *открытом мире*, согласно которому мы не делаем вывод о неверности утверждения исходя из его отсутствия (Reiter, 1978), в отличие от допущения о *закрытом мире*. Применительно к обнаружению объектов открытый мир означает, что

мы не отрицаем существование объекта при отсутствии соответствующего отклика детектора.

В контексте визуальных детекторов допущение о закрытом мире сталкивается с двумя проблемами:

- внешний вид объектов характеризуется высокой внутриклассовой изменчивостью, например автомобили и велосипеды в сценариях дорожного движения;
- набор объектов постоянно растет, например в сценариях дорожного движения появляются электровелосипеды и хOVERборды.

Если объект не обнаружен по первой причине, это означает, что детектор не смог смоделировать разнообразие внешнего вида объекта. Это может быть вызвано ограничениями обучающей выборки или бесконечными вариациями внешнего вида объекта. Если объект не обнаружен по второй причине, это означает, что класс объектов вообще не представлен в обучающей выборке. Однако тот факт, что набор данных не охватывает определенный класс, не означает, что этот класс не имеет отношения к детектору. Разработка системы, основанной на предпосылке, что классы, отсутствующие во время обучения, следует игнорировать, этически проблематична, например в приложениях для безопасности дорожного движения.

Системы безопасности дорожного движения часто основаны не на компьютерном зрении, а на сенсорах иного типа: лидары, радары, ультразвуковые эхолоты и т. д. Однако визуальные системы всегда будут представлять наибольший интерес, поскольку они наиболее близки к механизмам человеческого восприятия. Системы должны адаптироваться к среде, созданной для людей, и поскольку зрение является доминирующим чувством человека, обеспечивающим примерно 80 % нашего восприятия, обучения, познания и деятельности (Ripley, Politzer, 2010), таким системам необходимы визуальные сенсоры. В повседневной жизни мы постоянно руководствуемся визуальными средами. Например, правила дорожного движения определяют множество знаков и символов, которые визуальным образом направляют действия водителя. Кроме того, системы, которые делят свое рабочее пространство с людьми для взаимодействия и совместной работы, много выиграют от возможности визуального восприятия. Внешний вид человека может отчетливо сигнализировать о его поведении и намерениях, и люди активно используют это, общаясь, например, с помощью жестов. Наконец, чтобы иметь возможность предсказывать действия человека, необходимо также научиться предсказывать его восприятие, которое в основном основано на зрении.

9.1.4. Историческая справка

Многие из прежних подходов к визуальному отслеживанию восходят к проблемам дополненной реальности, где визуализация и восприятие объединяются в одной системе координат для создания среды смешанной реальности. Отслеживание важно здесь по двум причинам: относительное положение головы в сцене (Neumann et al., 1999) и положение движущихся объектов (Neumann, Park, 1998).

Далее мы кратко перечислим некоторые важные этапы в истории визуального отслеживания, а также ключевые работы и их авторов:

- 1981: алгоритм Лукаса–Канаде (Lucas, Kanade, 1981);
- 1984: чисто фазовый согласованный фильтр (Horner, Gianino, 1984);
- 1994: «Отслеживание – решаемая проблема» (из презентации Shi, Tomasi 1994);
- 1998: фильтры и эквивариантности, согласованные только по фазе (Felsberg, 1998);
- 2004: «Лукас–Канаде 20 лет спустя: объединяющая структура» (Baker, Matthews, 2004);
- 2007: проект MATRIS: L_1 -трекер и ковариация (Skoglund and Felsberg, 2007);
- 2013: новые задачи отслеживания ОТВ (Wu et al., 2013), VOT (Kristan et al., 2013) и генеративный комплексный трекер (Felsberg, 2013);
- 2014: победитель конкурса VOT2014 в области дискриминативного трекинга (DSST) (Danelljan et al., 2014; Kristan et al., 2015);
- 2018: дискриминативные холистические трекары победили в VOT2018 с точностью 75 % (Kristan et al., 2019);
- 2020: сиамские трекары и методы сегментации приобретают все большее значение (Kristan et al., 2020).

В последующих разделах мы подробно рассмотрим несколько основных событий, произошедших за этот 40-летний период, с особым акцентом на дискриминативные и основанные на обучении методы.

9.2. Методы на основе шаблонов

Прежде чем углубиться в методы, основанные на обучении, мы кратко обсудим другие методы, которые полностью основаны на исходной аннотированной ограничивающей рамке, часто называемой шаблоном.

9.2.1. Основы

В целом мы можем разделить методы на основе шаблонов на генеративные и дискриминативные. Эти методы концептуально показаны на блок-схемах на рис. 9.2.

Генеративный подход является наиболее часто используемым подходом на основе шаблонов. Здесь модель в основном представляет собой характерный *патч* (небольшой участок) изображения и обычно сама служит шаблоном. Локализация объекта в последующих кадрах достигается путем сопоставления модели с позициями-кандидатами в кадрах. Для сопоставления требуется мера расстояния между участками изображения, например L_2 -норма (метод наименьших квадратов) или L_1 -норма разности пикселей.

Поиск может быть выполнен путем полного перебора с применением высокоэффективных L_1 -методов (Skoglund, Felsberg, 2006), с использованием

эвристики для уменьшения окна поиска (например, динамические модели) или с помощью итерационных методов. Наиболее распространенный подход в первые годы был основан на L_2 -норме и итерациях, так называемом методе Лукаса–Канаде (Lucas, Kanade, 1981). Методы, основанные на этом подходе, подробно описаны в обзорной статье Бейкера и Мэтьюза (Baker, Matthews, 2004), поэтому мы ограничимся лишь обсуждением базового случая в конце раздела 9.9.

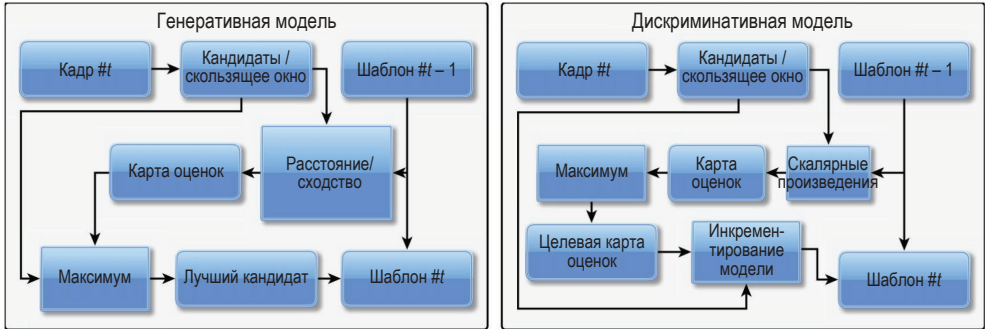


Рис. 9.2 ❖ Методы на основе шаблонов:
генеративный (слева, раздел 9.2.3)
и дискриминативный (справа, разделы 9.2.4 и 9.3.1)

Вместо этого мы сфокусируемся на дискриминативном подходе, когда модель двойственна по отношению к исходному шаблону. В самом простом случае *двойственность* (duality) понимается в терминах скалярного произведения в векторном пространстве. Предположим, что d векторов v_k порождают d -мерное векторное пространство. Мы не предполагаем, что эти векторы нормализованы или ортогональны. Двойственный базис распространен на d векторов \tilde{v}_k , таких что

$$\langle v_k | \tilde{v}_l \rangle = \begin{cases} 1, & \text{если } k = l \\ 0, & \text{если } k \neq l \end{cases} \quad \text{для } k, l \in \{1, \dots, d\}, \quad (9.1)$$

где $\langle | \rangle$ обозначает скалярное произведение. Заметим, что в случае ортонормированного базиса $\{e_k\}_{k=1\dots d}$ базисные векторы двойственны сами себе: $\tilde{e}_k = e_k$.

Если мы теперь перепишем вектор $v_k = \sum_{l=1}^d a_{kl} e_l$ с коэффициентами a_{kl} , то сразу увидим, что матрица A размера $d \times d$ с коэффициентами a_{kl} имеет ранг d и для нее существует обратная матрица. Без потери обобщения мы можем предположить, что $\{e_k\}_{k=1\dots d}$ является каноническим базисом \mathbb{R}^d , и, таким образом, $v_k, k = 1, \dots, d$, являются строками A . Если мы теперь применим циклический сдвиг к строкам A и умножим на A^{-1} , то получим уже не матрицу тождественности, а циклический сдвиг этой матрицы – мы успешно «отслежили» наш базисный вектор v_k .

Эту концепцию можно применить к любому векторному пространству, включая пространство функций или сигналов. Идея *чисто фазового согласо-*

ванного фильтра (phase-only matched filter) (Horner, Gianino, 1984) заключается в использовании двойственности шаблона для определения оценки общности (distinctive score). Этот показатель достигает своего максимума, если ограничивающая рамка расположена на шаблоне, и будет близок к нулю, если рамка снаружи.

9.2.2. Показатели качества модели

В процессе отслеживания мы находим ограничивающую рамку на основе некоторой оценочной функции; мы вернемся к маскам сегментации ниже в этом разделе. Чтобы количественно оценить, насколько успешно определено положение объекта, предсказанную ограничивающую рамку необходимо сравнить с истинной. Как в задаче VOT (Kristan et al., 2013), так и в ОТВ (Wu et al., 2013) в качестве критерия было выбрано пересечение-пересоединение (Everingham et al., 2010), или, исторически более правильно, индекс Жаккара J (Jaccard, 1912):

$$J = \frac{|R_G \cap R_P|}{|R_G \cup R_P|} = \frac{|R_G \cap R_P|}{|R_G| + |R_P| - |R_G \cap R_P|} = \left(\frac{|R_G| + |R_P|}{|R_G \cap R_P|} - 1 \right)^{-1}, \quad (9.2)$$

где R_G – это эталонная область, R_P – предсказанная область, а $|R|$ – площадь области R . Если две области не перекрываются, мы получаем $J = 0$, а если две области полностью совпадают, мы получаем $J = 1$. Если половина предсказанной области перекрывается с половиной эталонной области (типичная точность многих трекеров), мы получаем $J = \frac{1}{3}$. Заметим, что индекс Жаккара является смещенной мерой, которая склонна систематически завышать оценку размера ограничивающей рамки (Häger et al., 2018).

Индекс Жаккара измеряет точность отслеживания на кадр, и эти измерения можно накапливать несколькими способами. ОТВ предлагает рассчитать две кривые – кривую точности и кривую успеха (рис. 9.3). Обе рассчитываются аналогично ROC-кривой путем установки пороговых значений и вычисления соотношения кадров, которые проходят этот порог. Кривая точности получается путем установки порогового значения расстояния до центра рамки в диапазоне от 0 до максимального расстояния, а кривая успеха получается путем установки порогового значения индекса Жаккара в диапазоне от 0 до 1. Тогда интеграл под кривой является интегральной мерой точности. В качестве альтернативы можно использовать определенные пороговые значения, например расстояние 20 пикселей или перекрытие 50 % (Everingham et al., 2010).

В любом случае срыв отслеживания приводит лишь к снижению показателя точности, и это событие нельзя отличить от систематически низкой точности: при таких показателях точность и надежность сильно коррелируют (Kristan et al., 2016). Поэтому в задаче VOT стремятся разорвать корреляцию, используя две меры: точности и надежности. Первая измеряет средний индекс Жаккара в случае успешного отслеживания, а вторая измеряет частоту

неудачных попыток отслеживания (Kristan et al., 2013). Этот подход требует перезапуска отслеживания либо путем обнаружения сбоев и перезапуска отслеживания с использованием эталона, либо путем запуска нескольких попыток отслеживания из нескольких опорных точек в последовательности примерно в каждом 50-м кадре (Kristan et al., 2020). Точность накапливается по длине последовательности, а затем усредняется, при этом она взвешивается по соответствующей длине последовательности.

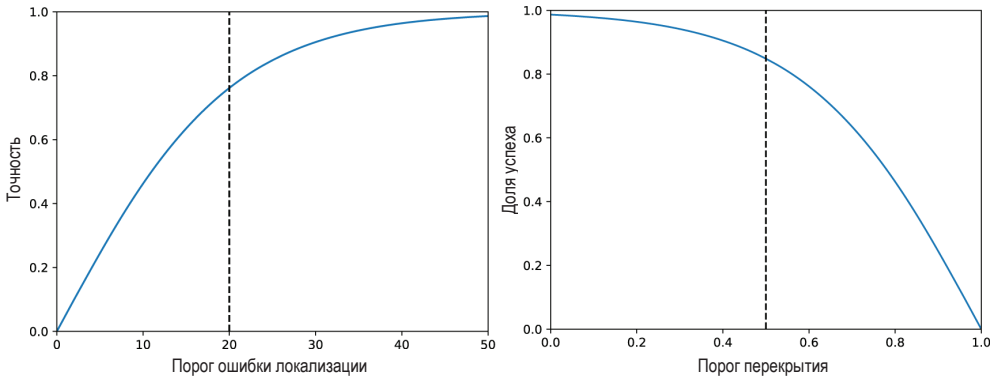


Рис. 9.3 ❖ Оценка отслеживания в ОТВ. Кривая точности (слева) и кривая успеха (справа). Ранжирование может быть выполнено по точности при пороге ошибки определения местоположения 20, степени успеха при пороге перекрытия 0,5 или по площади под кривой успеха (AUC)

При раздельном рассмотрении точности и надежности их обычно помещают на двухмерный график с надежностью по горизонтальной оси и точностью по вертикальной. В зависимости от назначения отслеживания разработчики могут отдать приоритет надежности или точности. Однако средние оценки не раскрывают полную картину и не говорят о том, имеет ли метод указанную среднюю точность на протяжении всего трека или изначально высокую точность, которая быстро снижается. Для многих приложений было бы интересно узнать, какую точность можно ожидать после определенной длины последовательности. Это достигается с помощью кривой ожидаемого среднего перекрытия, измеряющей среднее перекрытие как функцию длины последовательности. Окончательная оценка задачи VOT получается путем интегрирования этой кривой в диапазоне типичных длин последовательностей (Kristan et al., 2016).

В своей последней редакции (Kristan et al., 2020) задача VOT также предлагает вычислять показатели точности для масок сегментации вместо ограничивающих рамок. Такая необходимость возникла, поскольку методы отслеживания масочных сегментов достигли точности, аналогичной уровню точности аннотаций ограничивающей рамки. В отличие от индекса Жаккара для ограничивающих рамок, который можно вычислить как функцию параметров ограничивающей рамки, индекс Жаккара для масок сегментации требует подсчета пикселей в масках и на их пересечении.

9.2.3. Нормализованная кросс-корреляция

Один из старейших методов прогнозирования положения ограничительной рамки – сопоставление с шаблоном. Сопоставление выполняется с помощью генеративной модели m , которая получается из патча внутри ограничивающей рамки в исходном кадре. Во время сопоставления модель m сравнивается с патчами-кандидатами p в каждом последующем кадре. Соответственно, наиболее похожий патч определяет положение ограничивающей рамки в этом кадре (рис. 9.2 слева).

Моделью может служить фрагмент необработанного изображения в начальной ограничивающей рамке, но чаще предполагается, что модель m имеет нулевую DC-компоненту, т. е. вычитается среднее значение шаблона. Абсолютная интенсивность изображения часто зависит не от объекта, а от других факторов, т. е. освещения, теней, времени экспозиции и т. д., а удаление DC-компоненты повышает надежность согласования при изменении этих факторов. Однако в фотометрических применениях, например в последовательностях тепловых инфракрасных снимков, модель m может содержать DC-компоненту.

Для дальнейшего повышения надежности динамика интенсивности также обычно нормализуется с помощью дисперсий в шаблоне m и патче-кандидате p :

$$\sigma_m^2 = \frac{1}{|R|} \sum_{x,y} m(x,y)^2; \quad (9.3)$$

$$\sigma_p^2 = \frac{1}{|R|} \sum_{x,y} p(x,y)^2 - \left(\frac{1}{|R|} \sum_{x,y} p(x,y) \right)^2, \quad (9.4)$$

где $|R|$ – площадь области патча.

Патчи-кандидаты обычно выбираются скользящим окном по следующему кадру f или его части $p_{x,y}(r,s) = f(x+r, y+s)$. Оценка совпадения вычисляется как скалярное произведение между шаблоном и патчем. Эта комбинация скользящего окна и скалярного произведения дает нам корреляцию m и f :

$$c(x,y) = \langle p_{x,y} | m \rangle = \sum_{r,s} p_{x,y}(r,s) m(r,s) = \quad (9.5)$$

$$= \sum_{r,s} f(x+r, y+s) m(r,s) \stackrel{\text{def}}{=} (f \star m)(x,y). \quad (9.6)$$

Обратите внимание, что p не требует какой-либо DC-компенсации, потому что m не содержит DC, и, следовательно, скалярное произведение не содержит никакого вклада от среднего значения p .

Последующее деление на произведение стандартных отклонений дает *нормализованную кросс-корреляцию* (normalized cross correlation, NCC):

$$c_n(x,y) = \sigma_m^{-1} \sigma_{p_{x,y}}^{-1} c(x,y), \quad (9.7)$$

где $\sigma_{p_{x,y}}$ по-прежнему вычисляется в скользящем окне, что делает его функцией от (x, y) .

Корреляция $c(x, y)$ как таковая, т. е. без нормализации с помощью $\sigma_{p_{x,y}}$, может быть эффективно вычислена в пространстве Фурье (Bracewell, 1995)

$$C(u, v) \propto F(u, v) \circ \bar{M}(u, v), \quad (9.8)$$

где заглавные буквы – это преобразования Фурье соответствующих сигналов, обозначенных строчными символами, и \bar{M} комплексно сопряжена с M . Частотные координаты обозначаются через (u, v) , а оператор \circ является точечным произведением. Положение (\bar{x}, \bar{y}) вместе с максимальной нормализованной кросс-корреляцией (9.7) затем используется в качестве предсказания положения ограничивающей рамки.

Нормализованная кросс-корреляция используется не только с фиксированным, но и с адаптивным шаблоном. Единственное изменение в приведенной выше процедуре заключается в том, что шаблон из первого кадра используется только для инициализации модели m . После первого кадра локализованная ограничивающая рамка используется для обновления модели с помощью патча в этой ограничивающей рамке:

$$m \leftarrow (1 - \lambda)m + \lambda p_{\bar{x}, \bar{y}}, \quad (9.9)$$

где $\lambda \in (0, 1)$ обозначает коэффициент обновления. Если этот коэффициент выбран слишком большим, модель будет страдать от дрейфа (Matthews et al., 2004; Wang et al., 2016), а если слишком маленьким, то модель не сможет в достаточной мере адаптироваться к изменениям внешнего вида.

Заметим, что как в статическом, так и в адаптивном случае решение (9.7) идентично задаче наименьших квадратов для нормализованных патчей:

$$\min_{x,y} \|\sigma_{p_{x,y}}^{-1} p_{x,y} - \sigma_m^{-1} m\|^2 = \underbrace{\|\sigma_{p_{x,y}}^{-1} p_{x,y}\|^2 + \|\sigma_m^{-1} m\|^2}_{\text{постоянный член}} - 2 \max_{x,y} c_n(x, y). \quad (9.10)$$

Если задача наименьших квадратов решается итеративно методом градиентного спуска, мы получаем метод Лукаса–Канаде KLT (Lucas, Kanade, 1981). Этот метод является полностью локальным, т. е. он начинается с предыдущего местоположения (или местоположения, предсказанного какой-либо динамической моделью) и находит ближайшие локальные минимумы. Напротив, нормализованная кросс-корреляция определяет местонахождение глобального максимума во всем кадре f .

9.2.4. Чисто фазовый согласованный фильтр

Если мы присмотримся к вычислениям в пространстве Фурье (9.8), то может возникнуть идея выбрать M как

$$M(u, v) = \frac{F(u, v)}{|F|^2(u, v)} \quad (9.11)$$

так, что

$$C(u, v) = \frac{F(u, v) \circ \bar{F}(u, v)}{|F|^2(u, v)} = 1 \text{ и } c(x, y) = \delta(x, y). \quad (9.12)$$

Сдвиг f на (x_0, y_0) приводит к модуляции в пространстве Фурье (теорема о сдвиге) и, следовательно, к смещению Дирака в пространстве изображения. Таким образом, мы перешли к дискриминативной модели, так как фильтр не отображает внешний вид шаблона, а оценка выхода должна быть равна 1 для правильного перемещения и 0 для неправильного.

Для получения идеальной оценки требуется, чтобы все изображение было свободным от шума и смещалось вместе с патчем, содержащим цель. На практике это вряд ли возможно, и вычисленная оценка заменяется целевым показателем, соответствующим расположению с максимальным количеством баллов, для расчета новой модели (рис. 9.2 справа). Целевой показатель больше не является дираковским, поскольку спектр мощности $|F|^2(u, v)$ меняется между начальным кадром и последующими. По соображениям симметрии знаменатель в (9.11) необходимо изменить на $|F||F'|$, где F' – преобразование Фурье текущего кадра. Результирующий фильтр m является, строго говоря, нелинейным и также известен как *симметричный чисто фазовый согласованный фильтр* (symmetric phase-only matched filter, SPOMF) (Chen et al., 1994).

Название *чисто фазового согласованного фильтра* (phase-only matched filter, POMF) впервые предложено в работе (Horner, Gianino, 1984), где нелинейность устраняется за счет исключения $|F'|$ в знаменателе. Таким образом, эффективное согласование представляет собой корреляцию нового кадра с моделью m , имеющей постоянный амплитудный спектр

$$M(u, v) = \frac{F(u, v)}{|F|(u, v)}. \quad (9.13)$$

Этот фильтр больше не будет приводить к отклику Дирака (если только кадр не имеет спектра постоянной амплитуды). POMF, регуляризованный по амплитудному спектру текущего кадра, можно рассматривать в качестве дискриминаторного фильтра. Каждый коэффициент Фурье $C(u, v)$ умножается на соответствующий коэффициент спектра величин $|F|(u, v)$. На выходе получается не дираковская, а гладкая характеристика, форма которой получается из обратного преобразования Фурье амплитудного спектра f' . Эта регуляризация явно выражена в фильтре MOSSE, о котором говорится в разделе 9.3.1.

9.3. МЕТОДЫ ПОСЛЕДОВАТЕЛЬНОГО ОБУЧЕНИЯ

В этом разделе мы используем некоторые идеи из предыдущего раздела, регуляризацию отклика и инкрементное обновление модели, чтобы определить концепции дискриминативных корреляционных фильтров.

9.3.1. Фильтр MOSSE

Обсуждая POMF, мы отметили, что он имеет неявную регуляризацию отклика. Если эту регуляризацию сделать явной в виде целевой функции отклика c , мы приходим к концепции фильтра *минимальной выходной суммы квадратов ошибок* (minimum output sum of squared error, MOSSE) (Bolme et al., 2010):

$$\min_m \sum_{x,y} \left(\sum_{r,s} p_{x,y}(r,s) m(r,s) - c(x,y) \right)^2 = \min_m \|f \star m - c\|^2. \quad (9.14)$$

Обратите внимание, что хотя мы используем то же обозначение f , как и для всего кадра, мы будем подразумевать только локальное *окно поиска* (search window). Таким образом, мы получаем подход, который не является ни чисто локальным, как KLT, ни полностью глобальным, как NCC. Обычно окно поиска в два-три раза превышает размер ограничивающей рамки в обоих измерениях.

Фильтр MOSSE вычисляется в замкнутой форме с использованием эквивалентной формулы в пространстве Фурье:

$$\tilde{m} = \operatorname{argmin}_m \|f \star m - c\|^2 \quad (9.15)$$

$$= \operatorname{argmin}_m \|F \circ \overline{\mathcal{F}\{m\}} - C\|^2 \quad (9.16)$$

$$= \mathcal{F}^{-1} \left\{ \operatorname{argmin}_M \|F \circ \bar{M} - C\|^2 \right\}. \quad (9.17)$$

Это окончательное уравнение имеет решение (вывод см. в приложении к Bolme et al. (2010))

$$\tilde{m} = \mathcal{F}^{-1} \left\{ \frac{\bar{C} \circ F}{\bar{F} \circ F} \right\}. \quad (9.18)$$

Заметим, что эквивалентность достигается только для бесконечных областей. На практике извлеченный патч имеет конечный размер, и преобразование Фурье приводит к неявному периодическому повторению. Чтобы уменьшить влияние разрывов на границе патча, к нему применяется *окно Ханна* (косинусное окно у Bolme et al. (2010)).

Далее обратите внимание на сходство с (9.11), за исключением C . Как уже говорилось про POMF, мы эффективно применяем точечную регуляризующую функцию $C(u, v)$, и если мы выбираем $C = |F'|$, то получаем POMF. Однако для фильтра MOSSE наиболее распространенным вариантом оценки mAP с является функция Гаусса (не нормализованная) в точке (x_0, y_0) с фиксированной шириной

$$c(x, y) = \exp \left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2} \right). \quad (9.19)$$

Этот регуляризованный согласованный фильтр обычно обновляется в течение последовательности кадров, подобно обновлению NCC (9.9), см. также рис. 9.1. На каждом шаге целевая mAP-оценка $s(x, y)$ располагается на позиции максимальной оценки из предыдущей модели (рис. 9.2 справа).

Уравнение обновления MOSSE получается, если сначала рассмотреть задачу минимизации для нескольких аннотированных кадров (f_t, c_t) :

$$\min_m \sum_t \|f_t \star m - c_t\|^2. \quad (9.20)$$

В области Фурье это дает в силу линейности

$$\tilde{m} = \operatorname{argmin}_m \sum_t \|f_t \star m - c_t\|^2 \quad (9.21)$$

$$= \mathcal{F}^{-1} \left\{ \operatorname{argmin}_m \sum_t \|F_t \star \bar{M} - C_t\|^2 \right\} \quad (9.22)$$

с решением

$$\tilde{m} = \mathcal{F}^{-1} \left\{ \frac{\sum_t \bar{C}_t \circ F_t}{\sum_t \bar{F}_t \circ F_t} \right\}. \quad (9.23)$$

Если мы теперь предположим, что пары (f_t, c_t) поступают постепенно, нам нужно только разделить числитель и знаменатель, чтобы упорядочить обновление:

$$A_t = \bar{C}_t \circ F_t + A_{t-1}, \quad A_0 = 0 \quad (9.24)$$

и

$$B_t = \bar{F}_t \circ F_t + B_{t-1}, \quad B_0 = 0, \quad (9.25)$$

такой, что фильтр после временного шага t выглядит как

$$\tilde{m}_t = \mathcal{F}^{-1} \left\{ \frac{A_t}{B_t} \right\}. \quad (9.26)$$

Однако при увеличении количества кадров выражения для A_t и B_t подвержены неограниченному росту, чего мы хотим избежать по вычислительным соображениям. Кроме того, мы бы предпочли, чтобы модель последовательно уменьшала влияние очень старых образцов. Оба эффекта достигаются введением *коэффициента забывания* $1 - \eta$, где $0 < \eta < 1$, в уравнения обновления:

$$A_t = \eta \bar{C}_t \circ F_t + (1 - \eta) A_{t-1}, \quad A_1 = \bar{C}_1 \circ F_1; \quad (9.27)$$

$$B_t = \eta \bar{F}_t \circ F_t + (1 - \eta) B_{t-1}, \quad B_1 = \bar{F}_1 \circ F_1. \quad (9.28)$$

Обратите внимание, что, как и в обновленном NCC, фильтр с временным индексом t применяется к кадру $t + 1$.

В дополнение к неограниченному росту выражений знаменатель, близкий к нулю, тоже может вызвать вычислительные проблемы. Эта проблема была решена путем регуляризации коэффициентов фильтра со штрафом за их L_2 -норму (Henriques et al., 2012). Отсюда мы приходим к следующей задаче гребневой регрессии:

$$\min_M \sum_t \|F_t \circ \bar{M} - C_t\|^2 + \lambda \|M\|^2. \quad (9.29)$$

Решение представляет собой небольшую модификацию (9.23):

$$\tilde{m} = \mathcal{F}^{-1} \left\{ \frac{\sum_t \bar{C}_t \circ F_t}{\lambda + \sum_t \bar{F}_t \circ F_t} \right\}, \quad (9.30)$$

а для инкрементального случая мы только меняем (9.26) на

$$\tilde{m}_t = \mathcal{F}^{-1} \left\{ \frac{A_t}{\lambda + B_t} \right\} \quad (9.31)$$

и оставляем (9.27) и (9.28) без изменений.

9.3.2. Дискриминативные корреляционные фильтры

Фильтр MOSSE, рассмотренный в предыдущем разделе, действует непосредственно на данные, содержащиеся в изображении. Выводы также предполагают, что данные изображения являются скалярными (одноканальными), а обобщение на цветные изображения (многоканальные) нетривиально. Кроме того, нам может понадобиться использовать несколько признаков, извлеченных из изображения, вместо или в дополнение к чистым данным изображения. В этом случае также необходимо обрабатывать многоканальные входные данные, что приводит к обобщению (регуляризованного) фильтра MOSSE: *дискриминативному корреляционному фильтру* (discriminative correlation filter, DCF).

Обратите внимание, что для определения DCF мы меняем обозначения на символы, обычно используемые в машинном обучении. По сравнению с нотацией, основанной на обработке сигналов в фильтре MOSSE, мы применяем следующие сопоставления: $f \mapsto x$ для входных данных, $m \mapsto f$ для фильтра и $s \mapsto y$ для заданных выходных данных (Danelljan et al., 2015). Для входных данных с d каналами переформулируем (9.29) как минимизацию цели

$$\varepsilon_t(f) = \sum_{k=1}^t \alpha_k \|S_f(x_k) - y_k\|^2 + \lambda \sum_{l=1}^d \|f^l\|^2, \quad (9.32)$$

где мы определяем функцию вклада

$$S_f(x) = \sum_{l=1}^d x^l \star f^l \quad (9.33)$$

как многоканальную корреляцию между входом и фильтрами. Обратите внимание, что образцы (x_k, y_k) взвешиваются по $\alpha_k \geq 0$, что является дальнейшим обобщением по сравнению с введенным ранее коэффициентом забывания $(1 - \eta)$.

Решение задачи минимизации (9.32) получается путем подстановки функции вклада (9.33) и теоремы Парсеваля таким образом, что

$$\min_{f^n} \varepsilon_t(f) = \min_{f^n} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d x_k^l \star f^l - y_k \right\|^2 + \lambda \sum_{l=1}^d \|f^l\|^2 \quad (9.34)$$

$$= \min_{F^n} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d X_k^l \circ \bar{F}^l - Y_k \right\|^2 + \lambda \sum_{l=1}^d \|F^l\|^2 \quad (9.35)$$

для всех $n = 1 \dots d$. Необходимое условие минимума состоит в том, что частные производные по всем компонентам F^n и для всех n равны нулю:

$$0 = \frac{\partial \varepsilon_t}{\partial F^n} = \sum_{k=1}^t 2\alpha_k \left(\sum_{l=1}^d X_k^l \circ \bar{F}^l - Y_k \right) \bar{X}_k^n + 2\lambda \bar{F}^n \text{ для всех } u, v, n; \quad (9.36)$$

$$\mathbf{0} = \mathbf{X}^* \text{diag}(\alpha) \mathbf{X} \bar{\mathbf{F}} - \mathbf{X}^* \text{diag}(\alpha) \mathbf{Y} + \lambda \bar{\mathbf{F}}; \quad (9.37)$$

$$\bar{\mathbf{F}} = (\mathbf{X}^* \text{diag}(\alpha) \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^* \text{diag}(\alpha) \mathbf{Y}, \quad (9.38)$$

где второе равенство представляет собой матричную запись. Матрица \mathbf{X} содержит коэффициенты Фурье $X_k^l(u, v)$, $l = 1 \dots d$ как векторы-строки, \mathbf{X}^* – сопряженная транспонированная матрица, $\text{diag}(\alpha)$ – диагональная матрица с элементами α_k , $\bar{\mathbf{F}}$ содержит коэффициенты Фурье $\bar{F}^l(u, v)$, $l = 1 \dots d$ как вектор-столбец, а \mathbf{Y} – вектор-столбец, содержащий все Y_k . Исходные формулы можно найти в работе (Danelljan et al., 2015).

Решение (9.38) имеет несколько частных случаев. Регуляризованный фильтр MOSSE с инкрементным обновлением является частным случаем для $d = 1$ и α_k , образующим геометрическую прогрессию $\eta(1 - \eta)^{t-k}$. Другой частный случай получается, если мы рассматриваем только один-единственный образец. В этом случае \mathbf{X} – вектор-строка, а $\mathbf{X}\mathbf{X}^*$ – скаляр, так что

$$(\mathbf{X}^* \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^* \mathbf{Y} = \frac{\mathbf{X}^* \mathbf{Y}}{\mathbf{X}\mathbf{X}^* + \lambda} \quad (9.39)$$

и

$$\bar{F}^n = \frac{\bar{Y} \circ X^n}{\lambda + \sum_{l=1}^d \bar{X}^l \circ X^l}, \quad n = 1 \dots d. \quad (9.40)$$

Если мы обратимся к этому решению в случае нескольких образцов, то прием с обращением в (9.39) применим только в том случае, если $\mathbf{X}^* \text{diag}(\alpha)$ \mathbf{X} имеет единичный ранг. Без этого приема эффективное инкрементное обновление становится невозможным, поскольку нам пришлось бы инвертировать матрицу $d \times d$ для каждого коэффициента и каждого кадра. Существует обобщение приема с инверсией, известное как формула Шермана–Моррисона–Вудбери, и оно применялось к многоканальным DCF (Lukežić et al., 2018), но эмпирические результаты, похоже, указывают на то, что аппроксимация первого ранга работает достаточно хорошо (Danelljan et al., 2017). В этом приближении мы просто повторно используем модифицированные уравнения обновления (9.27) и (9.28):

$$A_t^n = \eta \bar{Y}_t \circ X_t^n + (1 - \eta) A_{t-1}^n, \quad A_1^n + \bar{Y}_1 \circ X_1^n, \quad n = 1 \dots d; \quad (9.41)$$

$$B_t = \eta \sum_{l=1}^d \bar{X}_t^l \circ X_t^l + (1 - \eta) B_{t-1}, \quad B_1 = \sum_{l=1}^d \bar{X}_1^l \circ X_1^l. \quad (9.42)$$

Подобно (9.31), мы вычисляем (векторный) фильтр как

$$\tilde{f}_t^n = \mathcal{F}^{-1} \left\{ \frac{A_t^n}{\lambda + B_t} \right\}, \quad n = 1 \dots d. \quad (9.43)$$

9.3.3. Подходящие признаки для DCF

С переходом от одноканальных фильтров MOSSE к DCF у нас появляется свобода выбора, какие функции используются для входной карты признаков x . При выборе признаков следует учитывать три основных компромисса: первый касается d , количества каналов в x , второй – пространственного разрешения или количества коэффициентов Фурье, а третий – размера временного окна в случае, если должны использоваться другие α_k , отличные от тех, что взяты из геометрической прогрессии.

Количество свободных параметров DCF зависит от размера карты признаков, d и пространственного разрешения, как следует из (9.36). Большее количество параметров, как и большее временное окно, внутреннее измерение в (9.38), требует больше вычислений. Кроме того, большее количество свободных параметров также требует более строгой регуляризации, поскольку количество исходных входных измерений, то есть кадр входного изображения, не зависит от размера карты признаков x .

Размер временного окна и пространственное разрешение будут рассмотрены в последующих разделах этой главы, а в этом разделе мы сосредоточимся на количестве каналов. Здесь основное внимание уделяется созданным вручную признакам: изученные и глубокие признаки также будут рассмотрены позже. Таким образом, задача, рассматриваемая в этом разделе, заключается в следующем: выбрать как можно меньше созданных вручную плотных признаков, чтобы получить высококачественный трекер DCF.

Поскольку подход DCF основан на отслеживании путем обнаружения, мы основываем выбор признаков на опыте из публикаций по обнаружению объ-

ектов. Если учитывать, что необработанные значения исходного изображения могут быть полезны в особых случаях (например, фотометрические данные, такие как тепловое ИК-излучение), структура и форма изображений, как правило, намного лучше представляются с использованием *гистограмм ориентированных градиентов* (HOG), предложенных для обнаружения человека (Dalal, Triggs, 2005). Признаки HOG – это особый способ представления данных об ориентации (Felsberg, 2018), как и плотные признаки SIFT (Lowe, 2004).

Большинство новейших методов отслеживания с использованием созданных вручную признаков основаны на признаках HOG, и нам не встречался какой-либо систематический анализ альтернативных вариантов, например на основе обобщенных формул (Felsberg, 2018). Кроме того, признаки, которые чаще используются для анализа текстуры, такие как бинарные локальные паттерны (Pietikäinen, Zhao, 2015), редко встречаются в методах отслеживания, по-видимому, из-за относительно низкой доли текстурированных объектов в наборах эталонных данных. Поэтому мы также предпочитаем использовать в данной главе признаки HOG, а за подробностями рекомендуем обратиться к первоисточнику (Felsberg, 2018).

Все рассмотренные до сих пор признаки игнорируют информацию о цвете в кадрах изображения. Опять же, из публикаций по обнаружению объектов мы узнаем, что было бы очень полезно использовать названия цветов (Khan et al., 2012). Deskриптор названия цвета вычисляется путем обучаемого мягкого присвоения из цветового пространства Lab одиннадцати названий цветов на английском языке (Van De Weijer et al., 2009). Обучение выполняется на слабо размеченных изображениях Google с использованием вероятностного латентно-семантического анализа (pLSA-bg):

$$P(w|d) = P(w|z)P(z|d), \quad (9.44)$$

где d обозначает распределение цвета в пространстве Lab, w – одно из одиннадцати названий цветов, а z – скрытая переменная. Названия цветов представлены 11-мерным *вектором мягкого присвоения* (soft-assignment vector), т. е. все компоненты находятся в диапазоне от 0 до 1, а вектор нормализован.

9.3.4. Отслеживание в масштабном пространстве

Трекеры DCF, представленные выше, по-прежнему страдают одним фундаментальным недостатком: если целевой объект перемещается по глубине, например ближе к камере, он меняет свой размер в плоскости изображения. Следовательно, необходимо менять не только положение ограничивающей рамки в текущем кадре, но и ее размер: нужно отслеживать объект, масштаб которого в общем случае постоянно меняется. Очевидно, внешний вид объекта также меняется в зависимости от масштаба или глубины, но существует тесная связь между масштабом и внешним видом, что позволяет нам связывать модели внешнего вида через масштаб – для этого введено понятие *масштабного пространства* (scale space) (Koenderink, 1984).

Ключевая идея масштабного пространства состоит в том, чтобы ввести в изображение третью (масштабную) ось, которая отражает степень размы-

тия, например вызванную ограниченной глубиной резкости камеры (Felsberg et al., 2005). Размытие обычно моделируется фильтрацией нижних частот, то есть уменьшением высокочастотных составляющих. В некотором масштабе s_0 частоты, превышающие половину частоты Найквиста $\pi/2$, ослабляются настолько, чтобы можно было выполнять понижающую дискретизацию с коэффициентом 2 без значительного размытия. При масштабе $2s_0$ мы можем уменьшить изображение с коэффициентом 4 и т. д., в конечном итоге построив *пирамиду масштаба* (рис. 9.4 слева).

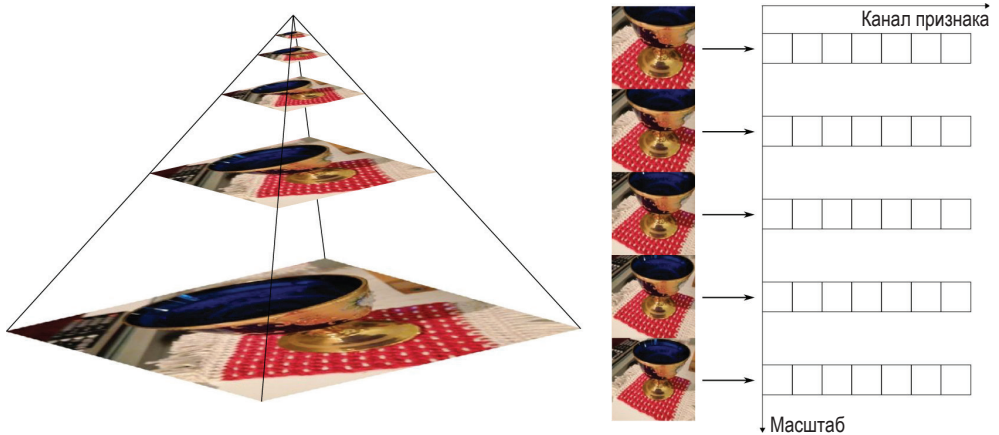


Рис. 9.4 ❖ Отслеживание в масштабном пространстве. Пирамида масштаба (слева) и отслеживание масштаба (справа). Ось масштаба соответствует пространственной координате глубины

Полезным побочным эффектом этой пирамиды является то, что уменьшенное пространственное разрешение в пирамиде соответствует уменьшенному размеру в плоскости изображения из-за увеличения расстояния от объекта до камеры. Таким образом, мы можем имитировать эффект увеличения глубины, переходя к более крупным масштабам в пирамиде, что позволяет нам включить оценку масштаба в отслеживание (Danelljan et al., 2017). Для этого процесса у нас есть три варианта:

- 1) применить трансляционный фильтр со множеством разрешений. Все соответствующие масштабы вычисляются параллельно, и выбирается лучшая оценка;
- 2) применить пространственный фильтр, включающий масштаб. DCF вычисляется в 3D, а не в 2D, а третья координата определяет размер объекта;
- 3) применить дискриминативное отслеживание масштаба. Вдоль оси масштаба применяется 1D-фильтр DCF для определения размера объекта.

Как показывает сравнение, первые два метода не только медленнее третьего, но даже не повышают точность (Danelljan et al., 2017). Поэтому в данном разделе мы рассмотрим только третий метод.

Основное предположение состоит в том, что масштаб изменяется медленнее, чем положение. Следовательно, мы можем разделить отслеживание

положения и масштаба, сначала оценив перемещение при постоянном масштабе с DCF, как было описано ранее, а затем оценив изменение масштаба. Масштабы, которые должны давать низкие оценки, должны генерироваться явным образом пирамидой масштаба области внутри ограничивающей рамки (рис. 9.4 справа), в отличие от случая трансляции, когда смещенные окна образованы сдвинутыми ограничивающими рамками. Затем соответствующие векторы признаков упорядочиваются для формирования карты признаков по координате масштаба и с использованием окна Ханна.

Псевдокод для дискриминативного отслеживания масштаба имеет следующий вид:

- Вход:
 - Изображение I_t
 - Предыдущее положение p_{t-1} и масштаб s_{t-1}
 - Модель переноса $A_{t-1,trans}, B_{t-1,trans}$
 - Модель масштаба $A_{t-1,scale}, B_{t-1,scale}$
- Выход:
 - Расчетное положение p_t и масштаб s_t
 - Обновленная модель переноса $A_{t,trans}, B_{t,trans}$
 - Обновленная модель масштаба $A_{t,scale}, B_{t,scale}$
- Оценка переноса:
 1. Извлечь образец $x_{t,trans}$ из изображения I_t для p_{t-1} и s_{t-1}
 2. Вычислить показатели корреляции $y_{t,trans}$, используя (9.33) с $f_{t-1,trans}$ согласно (9.43)
 3. Принять за p_t положение, максимизирующее $y_{t,trans}$
- Оценка масштаба:
 4. Извлечь образец $z_{t,scale}$ из изображения I_t для p_{t-1} и s_{t-1}
 5. Вычислить показатели корреляции $y_{t,scale}$, используя (9.33) с $f_{t-1,scale}$ согласно (9.43)
 6. Принять за s_t масштаб, максимизирующий $y_{t,scale}$
- Обновление модели:
 7. Извлечь образцы $x_{t,trans}$ и $x_{t,scale}$ из I_t для p_t и s_t
 8. Обновить модель переноса $A_{t,trans}, B_{t,trans}$, используя (9.41) и (9.42)
 9. Обновить модель масштаба $A_{t,scale}, B_{t,scale}$, используя (9.41) и (9.42).

Примечание. Масштабирование содержимого окна делает проблему с периодическим расширением очевидной, поскольку в этом процессе окно Ханна не масштабируется.

9.3.5. Пространственное и временное взвешивание

Несмотря на использование окна Ханна, неявное периодическое расширение патча, вызванное преобразованием Фурье, снижает дискриминативную способность фильтра. Признаки, характерные для интересующего объекта, вновь появляются вблизи патча и создают побочные пики в функции оценки. Наивная попытка противодействовать этим побочным пикам с помощью

увеличенных окон поиска терпит неудачу, поскольку большая часть рассматриваемого патча в таком случае состоит из фоновых пикселей. Фильтр склонен связывать эти фоновые пиксели с пиком функции, т. е. он изучает высокие коэффициенты в этих местах, и трекер будет «цепляться» за элементы фона, вместо того чтобы следовать за объектом (Danelljan et al., 2015).

Чтобы противодействовать этому эффекту, было предложено добавить к (9.32) пространственную регуляризацию, которая штрафует коэффициенты фильтра за пределами ограничивающей рамки (Danelljan et al., 2015):

$$\varepsilon_t(f) = \sum_{k=1}^t a_k \|S_f(x_k) - y_k\|^2 + \sum_{l=1}^d \|w \cdot f^l\|^2. \quad (9.45)$$

Однако требуемое поточечное умножение w и f в пространстве изображения становится сверткой в пространстве Фурье, что приводит к утрате основного преимущества DCF – вычислительной эффективности. Один из вариантов решения проблемы заключается в том, чтобы чередовать пространство изображения и пространство Фурье (Galoogahi et al., 2015), другой – выбрать весовую функцию w , которая имеет лишь несколько ненулевых коэффициентов Фурье. В последнем случае свертка в области Фурье не представляет проблемы, поскольку вычислительные затраты увеличиваются лишь незначительно (Danelljan et al., 2015).

Количество вычислений значительно сокращается заменой комплексных произведений Адамара (точечных) в (9.38) вещественными матричными произведениями. Начнем с системы уравнений

$$(\mathbf{X}^* \text{diag}(\alpha) \mathbf{X} + \lambda \mathbf{I}) \bar{\mathbf{F}} = \mathbf{X}^* \text{diag}(\alpha) \mathbf{Y} \quad (9.46)$$

и заменим \mathbf{X} на \mathbf{D} , $\bar{\mathbf{F}}$ на $\tilde{\mathbf{F}}$ и \mathbf{Y} на $\tilde{\mathbf{Y}}$, т. е. на соответствующие вещественные представления. Преобразование Фурье применительно к функции вещественных величин является эрмитово-симметричным, т. е. $P(-u, -v) = \bar{P}(u, v)$, а изометрическое отображение

$$(P(u, v), P(-u, -v)) \rightarrow (P(u, v) + P(-u, -v))/\sqrt{2}, (P(u, v) - P(-u, -v))/i\sqrt{2} \quad (9.47)$$

приводит к тому же решению, но с использованием вещественных, а не комплексных матриц. Если мы далее обобщим $\lambda \mathbf{I}$ как $\mathbf{W}^T \mathbf{W}$, представляя разреженную свертку, индуцированную пространственными весами, то получим

$$\left(\sum_{k=1}^t \alpha_k \mathbf{D}_k^T \mathbf{D}_k + \mathbf{W}^T \mathbf{W} \right) \tilde{\mathbf{F}} = \sum_{k=1}^t \alpha_k \mathbf{D}_k^T \tilde{\mathbf{Y}}_k. \quad (9.48)$$

Решим эту систему методом Гаусса–Зейделя, т. е. разобьем LHS на нижнюю треугольную и строго верхнюю треугольную части

$$\left(\sum_{k=1}^t \alpha_k \mathbf{D}_k^T \mathbf{D}_k + \mathbf{W}^T \mathbf{W} \right) \tilde{\mathbf{F}} = (\mathbf{L}_t + \mathbf{U}_t) \tilde{\mathbf{F}} = \sum_{k=1}^t \alpha_k \mathbf{D}_k^T \tilde{\mathbf{Y}}_k \quad (9.49)$$

и вычислим решение по итерациям

$$\mathbf{L}_t \tilde{\mathbf{F}}^{(j)} = \sum_{k=1}^t \alpha_k \mathbf{D}_k^T \tilde{\mathbf{Y}}_k - \mathbf{U}_t \tilde{\mathbf{F}}^{(j-1)}. \quad (9.50)$$

В частности, для w с несколькими ненулевыми коэффициентами Фурье этот метод быстро сходится и дает фильтры с малыми коэффициентами в окрестности. Заметим также, что этот метод позволяет избежать аппроксимативного решения в (9.42) и ограничения α_k на геометрическую прогрессию.

Полученная свобода выбора α_k использовалась для систематического уменьшения веса выборок, появляющихся в результате дрейфа трека от цели со временем (Danelljan et al., 2016). Веса α_k устанавливаются путем переопределения важности каждого кадра во время последующих выборок. Эта процедура помогает в неоднозначных случаях (например, при частичной окклюзии) и позволяет нам использовать всю доступную информацию.

Веса инициализируются из предыдущей информации, например из возраста выборки. Обычно используются априорные веса выборки ρ_k , обусловленные коэффициентом забывания $1 - \eta$ (9.9). Затем мы оптимизируем совместные потери

$$\min_{f, \alpha} \varepsilon_t(f, \alpha) + \frac{1}{\mu} \sum_{k=1}^t \frac{\alpha_k^2}{\rho_k} \quad (9.51)$$

$$\text{так, что } \alpha_k \geq 0, k = 1, \dots, t, \quad (9.52)$$

$$\sum_{k=1}^t \alpha_k = 1 \quad (9.53)$$

путем многократного обновления f с помощью итерации Гаусса–Зейделя (9.50) и α с помощью квадратичного программирования.

В разделе 9.4.2 будет представлен еще более эффективный способ оптимального использования всех доступных кадров, основанный на глубоких признаках.

9.4. Методы, основанные на глубоком обучении

Трекары DCF являются мощными методами, если применяются ранее введенные концепции регуляризации, масштабной адаптации и многоканальных функций оценки. Однако узким местом для дальнейшего улучшения является предопределенный слой признаков, созданных вручную. В этом разделе описывается переход к глубоким и адаптивным признакам и, наконец, сквозному обучению DCF.

9.4.1. Глубокие признаки в DCF

Использование созданных вручную признаков, описанных в разделе 9.3.3, было обусловлено традиционными методами обнаружения объектов. В ранних методах обнаружения объектов на основе глубоких нейросетей часто использовали прогнозирование регионов (Girshick et al., 2016), которые хуже подходят в качестве признаков для DCF. Поэтому в качестве первых попыток интегрировать глубокие признаки внешнего вида в DCF (Danelljan et al., 2016) вместо сетей-детекторов использовали базовые решения из области классификации изображений, например imagenet-vgg-m-2048 / CNN-M-2048 (Chatfield et al., 2014).

При рассмотрении базовой сети для создания карты признаков в качестве входных данных, подаваемых в DCF, возникает очевидный вопрос, какой слой использовать. Кандидатами являются сверточные слои, например с первого по пятый в imagenet-vgg-m-2048. Самые верхние слои характеризуются высоким пространственным разрешением и небольшим количеством каналов (например, 109×109 , 96 каналов для первого слоя в imagenet-vgg-m-2048), а самые глубокие слои – низким пространственным разрешением и большим количеством каналов (например, 13×13 , 512 каналов для пятого слоя в imagenet-vgg-m-2048). Как правило, более глубокие слои компенсируют большее количество каналов меньшим пространственным разрешением, что является неизбежным следствием постоянства соотношения между неопределенностью в пространстве изображения и пространстве признаков (Felsberg, 2009).

В то время как большее количество каналов признаков обеспечивают лучшую дискриминативную способность, уменьшенное пространственное разрешение может привести к ухудшению точности. С точки зрения VOT-критериев робастности и точности предполагается, что DCF на основе неглубоких признаков будет характеризоваться высокой точностью и низкой робастностью, тогда как DCF на основе глубоких признаков будет характеризоваться низкой точностью и высокой робастностью. Это также неявно подтверждается экспериментами с imagenet-vgg-m-2048, где первый и пятый уровни являются локальными максимумами точности перекрытия ОТВ-50 (Danelljan et al., 2016) – меры, которая объединяет точность и робастность (Wu et al. др., 2013).

Разумеется, нам хотелось бы объединить обе карты признаков, чтобы использовать как высокую пространственную точность неглубоких признаков, так и робастность глубоких. Однако различные пространственные разрешения потребуют либо повышающей дискретизации глубоких признаков, что невозможно с вычислительной точки зрения, либо понижающей дискретизации неглубоких, что снизит положительный эффект в отношении точности. Идеальным случаем было бы, если бы все карты признаков были непрерывными функциями и дискретизация вообще не требовалась. Но как представить непрерывную карту признаков в цифровом компьютере?

Решение кроется в изометрических преобразованиях непрерывных сигналов в дискретные спектры, такие как ряд Фурье или другие разложения

в ряд (Danelljan et al., 2016), – это трекер C-COT. Чтобы упростить изложение, мы сосредоточимся на рядах Фурье с конечным числом коэффициентов, но метод обобщается на все изометрические преобразования. Начнем с пересмотра расчета mAP-оценки с использованием корреляций карты признаков и набора фильтров (9.33). Если мы предположим, что эта корреляция выполняется в непрерывной области между непрерывной картой признаков и множеством непрерывных фильтров

$$S_f(x) = \sum_{l=1}^d f^l \star J_l\{x^l\}, \quad (9.54)$$

где $J_l\{x^l\}$ – гипотетическое отображение дискретной карты признаков x^l в непрерывную область, мы можем легко объединить карты признаков с различным пространственным разрешением.

Если мы теперь переместим всю систему в пространство Фурье, – нам все равно придется сделать это для вычисления решений (9.43), – непрерывная корреляция станет поточечным произведением дискретных коэффициентов. Периодическая непрерывная функция имеет бесконечно много дискретных коэффициентов Фурье, но мы также знаем, что усечение последовательности коэффициентов приведет к оптимальной аппроксимации в смысле L_2 (кстати, это особое свойство ряда Фурье). Аппроксимация mAP-оценки оптимальным для L_2 способом идеально соответствует идее MOSSE о минимальной L_2 -ошибке вывода.

В итоге это означает, что, используя решение (9.50) (или более эффективную формулировку с применением метода сопряженных градиентов), мы приобретаем все инструменты для слияния карт признаков с различным разрешением – в отличие от подхода, основанного на ADMM (Galoogahi et al. al., 2015), который переключается между пространством изображения и пространством Фурье. В дополнение к объединению первого и пятого слоев и необработанных входных данных (224×224 , RGB) непрерывная модель допускает субпиксельную локализацию во время логического вывода и обучения, что улучшает результаты даже больше, чем повышение дискретизации до самого высокого разрешения. Заметим в этом контексте, что для целевой функции у известна ее аналитическая форма (уравнение 9.19), поэтому мы можем точно вычислить коэффициенты Фурье. Обратите внимание, что этот подход также можно использовать для отслеживания характерных точек в качестве альтернативы KLT (Lucas, Kanade, 1981).

После перехода к многослойным признакам внешнего вида можно интегрировать другие модальности. Ранее игнорируемый тип признаков в DCF-трекерах – это признаки движения, которые теперь можно объединять с признаками внешнего вида (Danelljan et al., 2019). Признаки внешнего вида, применяемые в этой работе, взяты из сети imagenet-vgg-verydeep-16/ConvNet C (Simonyan, Zisserman, 2015) с 13 сверточными слоями и глубокими признаками движения (Chéron et al., 2015), вычисленными с трехканального входа (вектор оптического потока и его величина), состоящего из пяти сверточных слоев. Для отслеживания наилучшие результаты дает объединение

слоев внешнего вида 4 (128 каналов, страйд 2) и 13 (512 каналов, страйд 16) и пятого слоя движения (384 канала, страйд 16) (Danelljan et al., 2019).

9.4.2. Адаптивные глубокие признаки

Применение интеграции нескольких карт признаков для DCF-трекеров приводит к значительному улучшению результатов по сравнению с предшествующими методами как с использованием созданных вручную признаков, так и с однослойными глубокими признаками, но также вызывает увеличение количества параметров в модели фильтра. Большое количество параметров проблематично в двух отношениях: во-первых, от него напрямую зависит объем вычислений. Во-вторых, количество обучающих данных ограничено, а количество параметров легко выходит за пределы размерности входных данных, что приводит к переобучению.

Например, было обнаружено, что трекер C-COT, использующий признаки imagenet-vgg-m-2048, при последовательном обучении набирает до 800 000 параметров (Danelljan et al., 2017). Чтобы решить возникающие в результате проблемы сложности сети и нехватки данных для обучения, был предложен трекер ECO, основанный на эффективных операторах корреляции и дискриминативно изучающий отображение в пространстве признаков более низкой размерности (Danelljan et al., 2017). Это достигается за счет совместной минимизации ошибки классификации, что приводит к сокращению количества параметров модели на 80 %.

Ключевая идея состоит в том, чтобы ввести некоторый оператор проекции P , который перед вычислением оценки отображает d каналов непрерывных признаков в (9.54) в $c \ll d$ измерений. На этот оператор не влияет преобразование Фурье. Добавив регуляризатор для L_2 -нормы всех коэффициентов оператора P , мы переформулируем (9.45) в виде

$$\varepsilon_t(f) = \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^c f^l \star p_l^T J\{x_k\} - y_k \right\|^2 + \sum_{l=1}^c \|w \cdot f^l\|^2 + \lambda \|P\|_F^2, \quad (9.55)$$

где p_l – векторы-столбцы в P , а последний член – норма Фробениуса. Эти потери минимизируются с помощью метода Гаусса–Ньютона, полученного путем линеаризации остатков. За дополнительной информацией мы отсылаем читателей к оригинальной публикации (Danelljan et al., 2017).

Несмотря на достигнутое улучшение соотношения параметров и количества обучающих данных, необходимое разнообразие обучающих данных все равно потребует достаточно большой выборки, что приводит к значительной вычислительной нагрузке. Кроме того, размер памяти ограничен, что приводит к сокращению временного горизонта последовательного обучения, а отбрасывание старых выборок приводит к переобучению на последних кадрах.

Чтобы избежать этого переобучения, сумма по всем выборкам от $k = 1$ до t в (9.55) заменяется математическим ожиданием по совместному распределению $p(x, y)$:

$$\varepsilon_t(f) = \mathbb{E}_{p(x,y)} \left\{ \left\| \sum_{l=1}^c f^l \star p_l^T J\{x\} - y \right\|^2 \right\} + \sum_{l=1}^c \|w \cdot f^l\|^2 + \lambda \|P\|_F^2. \quad (9.56)$$

Неявное использование исходной суммы означает предположение, что из $p(x, y)$ взяты t выборки и их достаточно много, чтобы точно аппроксимировать $p(x, y)$, однако это может быть неправильным предположением.

Для получения компактного и разнообразного представления обучающих данных ЕСО-трекер моделирует $p(x, y)$ как смесь моделей Гаусса. Сочетание двух улучшений по сравнению с C-COT не только повышает скорость, но и улучшает точность на 13,3 % (Danelljan et al., 2017).

Подход, применяемый в трекерах C-COT и ECO, не ограничивается признаками imagenet-vgg-m-2048, но, что несколько неожиданно, использование более мощных базовых моделей, таких как GoogLeNet (Szegedy et al., 2015) или ResNet-50 (He et al. al., 2016), не улучшает точность (Bhat et al., 2018). Чтобы эффективно раскрыть возможности глубокого отслеживания, карты признаков не должны объединяться на уровне входных данных средства отслеживания, как это сделано в (9.54). Вместо этого лучший эффект дает обучение глубоких и неглубоких моделей независимо друг от друга и с разной шириной целевой функции, а также применение взвешенного слияния раздельно обученных трекеров.

Как отмечалось в разделе 9.4.1, прогнозы, основанные на карте глубоких признаков $\hat{y}_d = S_f(x_d)$, характеризуются высокой надежностью, но худшей локализацией, а прогнозам, основанным на карте неглубоких признаков $\hat{y}_s = S_f(x_s)$, присуща высокая точность локализации, но они легко теряют цель при изменении внешнего вида. Два прогноза можно комбинировать различными способами, но адаптивно взвешенная сумма

$$\hat{y}_\beta(p) = \beta_d \hat{y}_d(p) + \beta_s \hat{y}_s(p) \quad (9.57)$$

уже приводит к значительному улучшению по сравнению с трекером ECO. На каждом временном шаге адаптивные веса вычисляются путем решения задачи квадратичного программирования

$$\min_{\xi, b} -\xi + \mu(\beta_d^2 + \beta_s^2) \quad (9.58)$$

$$\text{так, что } \beta_d + \beta_s = 1, \beta_d \geq 0, \beta_s \geq 0, \quad (9.59)$$

$$\hat{y}_\beta(p^*) - \xi \Delta(p^* - p) \geq \hat{y}_\beta(p), \forall p, \quad (9.60)$$

где p^* – потенциальное целевое состояние (например, позиция), которое выбирается из локальных максимумов в \hat{y}_d и \hat{y}_s , μ – параметр регуляризации, а $\Delta(p) = 1 - \exp(-4|p|^2/s)$, где s является размером цели.

9.4.3. DCF сквозного обучения

Подходы, описанные в предыдущем разделе, добавляют адаптацию признаков в DCF-трекеры с глубокой сетью. Базовая глубокая сеть обучается в авто-

номном режиме на данных классификации изображений, трекер обучается в режиме реального времени с использованием текущей последовательности, а процедура адаптации представляет собой инженерную задачу оптимизации. Если мы хотим вместо этого сосредоточиться на обучении для подключения базовой модели к трекеру, нам нужно использовать обучающие данные, относящиеся к отслеживанию. Основное различие между адаптацией и обучением заключается в данных, используемых для оптимизации: в отличие от адаптации, обучение пытается обобщить обучающие данные на рабочие данные во время вывода.

Ключевая проблема, решаемая адаптивной комбинацией mAP-оценок в (9.57), состоит в том, чтобы генерировать унимодальные функции оценок с максимумом, сосредоточенным внутри ограничивающей рамки. Объединение функций оценки из неглубоких и глубоких признаков является хорошо работающим эвристическим решением, но подход машинного обучения интересен тем, что напрямую решает саму задачу. Следовательно, задача сводится к тому, чтобы на обучающих данных отслеживания изучить предиктор меры точности – индекс Жаккара. Вместо того чтобы выбирать все локальные максимумы и оптимизировать веса, предсказание индекса Жаккара позволяет выбрать ограничивающую рамку-кандидата с наибольшим перекрытием.

Эта идея максимизации перекрытия используется в трекере ATOM (Danieljan et al., 2019), который сочетает в себе обучаемый онлайн-классификатор (похожий на DCF) с индексным регрессором Жаккара, который обучается на аннотированных данных отслеживания (рис. 9.5). Очевидно, что регрессорная сеть должна учитывать конкретную цель, заданную в исходной (эталонной) системе координат. Это достигается путем модуляции вектора коэффициентов текущего кадра соответствующим вектором из опорного кадра перед подачей его в предиктор.

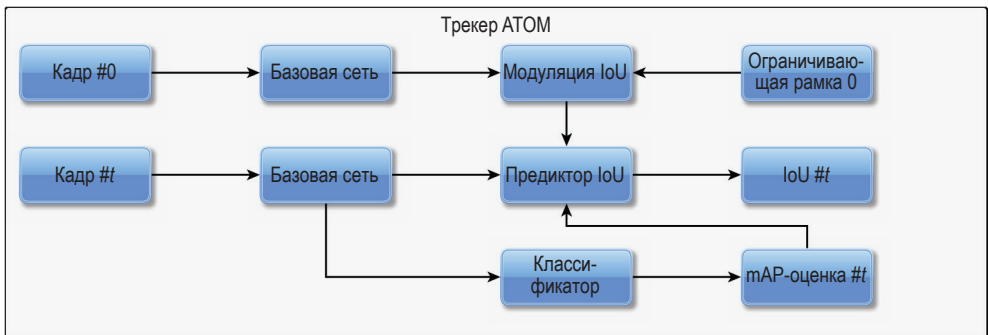


Рис. 9.5 ❖ Трекер ATOM состоит из ветви классификации и ветви регрессии для индекса Жаккара (пересечение над объединением)

Две ветви, для системы отсчета и текущей системы координат, очень похожи по структуре. Основное отличие состоит в том, что ветвь тестирования для текущего кадра не имеет доступа к ограничивающей рамке, а вместо этого использует начальную оценку из классификатора, подобного DCF. Весь регрессор подвергается сквозному обучению, при этом классификатор

обучается не с использованием потерь с обратным распространением, а с использованием критерия MOSSE. Таким образом, классификатор действует как фиксированный вход в сеть регрессора, которая имеет двухветвенную архитектуру, напоминающую сиамский трекер (Zhu et al., 2018). Подробное описание сетевой модели регрессора индекса Жаккара можно найти в статье (Danelljan et al., 2019) или репозитории кода (<https://github.com/visionml/pytracking>).

Классификатор в основном аналогичен классификатору в ECO-трекере, с первым уровнем, который сокращает количество каналов признаков до 64 (оператор проекции), и вторым уровнем с ядром 4×4 . Из-за ограниченного размера пространственного ядра последовательная оптимизация теперь выполняется не в пространстве Фурье, а непосредственно в пространстве изображения. Одним из преимуществ этого изменения является то, что к выходным данным свертки можно добавить нелинейную функцию активации, в данном случае параметрический экспоненциально-линейный блок.

В дальнейшем развитии трекера АТОМ, которое можно рассматривать как слияние DCF-подобных подходов и сиамских трекеров (Чжу и др., 2018), цель MOSSE исключена, и потери вместо этого изучаются на основе данных (Bhat et al., 2019). Этот метод под названием DiMP можно рассматривать как следующий шаг дискриминативного обучения, где дискриминационная способность оценивается не по L_2 -расстоянию, а по некоторой функции расстояния, зависящей от данных. Это также повышает гибкость представления результатов отслеживания: вместо параметрического ограничивающего прямоугольника можно легко перейти к маскам сегментации.

9.5. ПЕРЕХОД ОТ ОТСЛЕЖИВАНИЯ К СЕГМЕНТАЦИИ

Параметрическая модель ограничивающей рамки неявно подразумевает формулировку целевой функции как функции Гаусса с центром в истинном положении и масштабе. Помимо смещения, упомянутого в разделе 9.2.2, ограничивающая рамка также страдает от присущей ей неточности для объектов, форма которых отличается от прямоугольной. Борьба с этой проблемой начинается с аннотирования, когда общая точность повышается за счет аннотирования масок сегментации и автоматической подгонки ограничивающих рамок (Vojír, Matas, 2017; Kristan et al., 2016).

9.5.1. Сегментация видеообъектов

С 2020 года задача VOT оценивает индекс Жаккара по маскам сегментации, а не ограничивающим рамкам (Kristan et al., 2020). Это означает, что отслеживание визуальных объектов стало похоже на *сегментацию видеообъектов* (video object segmentation, VOS) для случая с одним экземпляром (Perazzi et al., 2016). Задача в VOS состоит в том, чтобы классифицировать каждый пиксель в кадре видео либо как фон, либо как часть целевого объекта. Методы

VOS обучаются в автономном режиме на аннотированных видеопоследовательностях и оцениваются на тестовых последовательностях с частично новыми (незнакомыми) классами объектов.

Как и в случае с VOT, при оценке методов VOS используется индекс Жаккара (раздел 9.2.2). Однако два прогноза могут иметь одинаковый индекс Жаккара, но совершенно разные формы, которые можно оценить только с помощью меры контура. В тестовой задаче DAVIS (Perazzi et al., 2016) для этой цели используется \mathcal{F} -мера контура, определяемая как

$$\mathcal{F} = 2 \frac{P \cdot R}{P + R}, \quad (9.61)$$

где P – точность (precision), а R – полнота отклика (recall). Для контуров точность вычисляется как доля предсказанных пикселей контура, которые действительно являются таковыми, а полнота вычисляется как доля пикселей истинного контура, которые являются предсказанными пикселями контура. Обычно \mathcal{F} -мера вычисляется приближенно с использованием морфологических операторов (Perazzi et al., 2016).

В случае обучения с частичным участием учителя (Perazzi et al., 2016) первый кадр тестовой последовательности аннотируется маской сегментации, аналогично заданию VOT2020. Следовательно, метод VOS должен адаптироваться к этому единственному аннотированному образцу, решая задачу однократного обучения. В случае обучения без учителя (Perazzi et al., 2016) аннотация не предоставляется, но предполагается, что целевой объект неявно определяется его движением относительно фона. Наконец, существует также сценарий множественных объектов с частичным обучением, когда в начальном кадре аннотируют несколько объектов (Perazzi et al., 2017). Эта задача также тесно связана с проблемой *сегментации экземпляров видео* (video instance segmentation, VIS), где, как и при обнаружении объектов на неподвижных изображениях, необходимо сегментировать все известные объекты по всей последовательности (Yang et al., 2019).

В оставшейся части этой главы мы сосредоточимся на проблеме VOS с частичным обучением для случая с одним экземпляром, поскольку она наиболее близка к проблеме VOT с аннотацией маски сегментации. В определенном смысле этот переход к маскам сегментации можно рассматривать как конечную точку развития методов отслеживания ограничивающей рамки, аналогично тому, как трекер ATOM завершает серию DCF на основе преобразования Фурье, а трекер DiMP завершает серию DCF, основанных на MOSSE. Но прежде чем рассматривать дискриминативные методы VOS, мы сначала изучим генеративный подход, чтобы прояснить разницу между методами VOS.

9.5.2. Генеративный метод VOS

Несмотря на то что область VOS довольно молода даже по меркам компьютерного зрения, в ней уже появилось множество различных подходов

к глубокому обучению. Мы не станем давать здесь обзор методов глубокого обучения для VOS и сошлемся на недавние обзорные статьи по этой теме, например (Yao et al., 2019). Однако стоит заметить, что многие методы предусматривают масштабную тонкую настройку сети, используя аннотацию первого кадра тестовых последовательностей. Это решение не подходит для практического применения, поскольку означает, что видео не может обрабатываться на лету, а задержка между входом и откликом модели в лучшем случае составляет несколько минут.

Один из подходов, позволяющих избежать столь прямолинейной и трудоемкой настройки, основан на *генеративной модели внешнего вида* (a generative appearance model, AGAME) (Johndander et al., 2019). Данный метод оценивает параметры *гауссовой смешанной модели* (Gaussian mixture model, GMM) и извлекает глубокие признаки из

$$p(\mathbf{x}) = \sum_{k=1}^K p(z = k)p(\mathbf{x}|z = k), \quad (9.62)$$

где $p(\mathbf{x}|z = k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Подразумевается единый prior $1/K$ для всех компонент k , которые находятся либо на заднем, либо на переднем плане. Каждый из этих двух классов разбивается на основную моду и моду сложных случаев, т. е. всего мы получаем четыре компонента.

Параметры GMM θ , т. е. средние $\boldsymbol{\mu}_k$ и дисперсии $\boldsymbol{\Sigma}_k$, инициализируются из начального кадра. Во время развертывания модели, т. е. последующих кадров I^i , когда доступны только оценки принадлежности к классу, для обновления модели используются следующие присвоения:

$$\alpha_0^i = 1 - \hat{y}(I^i, \theta^{i-1}, \Phi); \quad (9.63)$$

$$\alpha_1^i = \hat{y}(I^i, \theta^{i-1}, \Phi); \quad (9.64)$$

$$\alpha_2^i = \max(0, \alpha_0^i - p(z^i = 0|\mathbf{x}^i, \boldsymbol{\mu}_0^i, \boldsymbol{\Sigma}_0^i)); \quad (9.65)$$

$$\alpha_3^i = \max(0, \alpha_1^i - p(z^i = 1|\mathbf{x}^i, \boldsymbol{\mu}_1^i, 1)). \quad (9.66)$$

Здесь Φ обозначает параметры сетей слияния и прогнозирования.

Грубые прогнозы \hat{y} вычисляются путем слияния выхода распространения маски и оценки компонента (логарифмические вероятности $\ln p(z = k)p(\mathbf{x}|z = k)$):

$$s_k^i = -\frac{\ln|\boldsymbol{\Sigma}_k^{i-1}| + (\mathbf{x}^i - \boldsymbol{\mu}_k^{i-1})^T (\boldsymbol{\Sigma}_k^{i-1})^{-1} (\mathbf{x}^i - \boldsymbol{\mu}_k^{i-1})}{2}, \quad (9.67)$$

в результате получаются значения для компонентов 0 и 1. Вероятности в случаях 2 и 3 вычисляются с использованием мягкого максимума соответствующих оценок. Четыре мягких значения используются для вычисления обновлений среднего значения и дисперсии как взвешенных первого и второго моментов. Затем обновления передаются в скользящее среднее, аналогичное (9.9).

Во время вывода прогнозы уточняются модулем повышающей дискретизации для достижения полного разрешения. Уровень точности алгоритма AGAME делает его применимым для полуавтоматического аннотирования в задаче RGBT VOT2019 после дополнительных проверок правильности (Berg et al., 2019).

9.5.3. Дискриминативный метод VOS

Подобия в случаях 2 и 3 в предыдущем разделе «обеспечивают кодирование дискриминативной маски» (Johnander et al., 2019), что сразу наводит на мысль, можно ли заменить GMM дискриминативной моделью, т. е. выделенной целевой моделью, подобной той, что в трекаре АТОМ. Подобно GMM в AGAME, эта целевая модель должна адаптироваться к внешнему виду цели путем надлежащего обновления, чтобы максимизировать разделение переднего плана и фона. Ожидается, что фокусировка на дискриминативной способности, а не на всем распределении, приведет к созданию более легкого и быстрого алгоритма по сравнению с AGAME.

Эту идею подтверждает дискриминативный метод VOS FRTM (Robinson et al., 2020), который также заменяет распространение маски, используемое в AGAME, на пространственно-временную согласованность. Целевая модель напоминает трекаре АТОМ

$$s = D(\mathbf{x}; \mathbf{w}) = \mathbf{w}_2 * (\mathbf{w}_1 * \mathbf{x}) \quad (9.68)$$

и последовательно обучается с использованием взвешенного MOSSE-критерия

$$\mathcal{L}_D(\mathbf{w}) = \sum_k \gamma_k \|\mathbf{v}_k \circ (\mathbf{y}_k - U(D(\mathbf{x}_k)))\|^2 + \sum_j \lambda_j \|\mathbf{w}_j\|^2, \quad (9.69)$$

где \mathbf{v}_k – балансирующий вес для увеличения значимости небольших целевых объектов, U выполняет билинейную повышающую дискретизацию, а γ_k – веса для контроля влияния различных образцов в наборе данных.

В последовательном обучении, как и в методах отслеживания, используется временное окно. Данные из этого окна хранятся в памяти M .

Псевдокод для обучения целевой модели выглядит следующим образом:

1. Инициализировать память M , обучить дискриминационную целевую модель D .
2. Извлечь признаки x из следующего кадра.
3. Применить целевую модель и сгенерировать грубые mAP-оценки s .
4. Улучшить s до целевой маски u с помощью уточняющей сети.
5. Обновить M новым образцом (x, u, γ) .
6. Повторно оптимизировать дискриминативную целевую модель с помощью M в каждом восьмом кадре.
7. Вернуться к шагу 2.

Уточняющая модель, использованная на шаге 4, обучается в автономном режиме на этапе обучения VOS. Назначения грубой оценки s и уточненной

маски у аналогичны AGAME, но AGAME использует грубый прогноз для обновления модели, тогда как FRTM использует уточненный прогноз для обучения, предварительно применяя билинейную повышающую дискретизацию. Благодаря этим изменениям FRTM становится очень быстродействующим и позволяет использовать алгоритмы VOS в реальном времени.

Подобно переходу от ATOM к DiMP с обобщением потерь, целевая модель VOS также может быть оптимизирована в отношении представления, отличного от предсказания грубой сегментации. Предположим, что целевая модель (9.68) создает некоторую карту признаков вместо карты грубой сегментации и что уточненная сегментация создается декодером, а не модулем повышающей дискретизации. В этом случае потери для целевой модели могут быть вычислены в области карты признаков путем кодирования маски сегментации и вычисления взвешенной ошибки L_2 между закодированной сегментацией и выходными данными целевой модели (Bhat et al., 2020).

Этот подход связывает обучение VOS с метрическим обучением. Другими интересными вопросами являются обобщение на VOS с обучением без учителя, VOS с несколькими экземплярами объектов и VIS.

9.6. Выводы

Развитие методики от простого сопоставления шаблонов до сегментации видео, описанное в этой главе, является одним из краеугольных камней для многих приложений компьютерного зрения. Анализ видеоряда, визуальное наблюдение, дистанционное слежение, *дополненная реальность* (augmented reality, AR), визуальное управление роботами и автономные транспортные средства – это всего лишь несколько основных областей применения разработанных методов. Большинство этих приложений требуют обработки в реальном времени, надежных моделей и точных прогнозов.

Например, AR требует обработки видео в реальном времени с малой задержкой, чтобы уменьшить дискомфорт от AR¹. Безмаркерные методы дополненной реальности в значительной степени зависят от визуального отслеживания (Chandaria et al., 2007). Совмещение ракурса камеры и виртуального положения также можно использовать для управления беспилотными транспортными средствами. Например, виртуальное привязывание можно реализовать с помощью трекера DCF на кадрах RGB с дрона (Häger et al., 2016).

Входные данные, обсуждаемые в этой главе, были в основном ограничены последовательностями RGB, но аналогичные методы можно применять и к другим спектральным диапазонам или данным о глубине. Например, системы безопасности поездов на основе тепловизионных инфракрасных камер могут смягчать последствия столкновений с крупными животными,

¹ Укачивание, тошнота и головная боль, возникающие вследствие почти неосознанной, но неестественно большой задержки зрительных образов относительно остальных органов чувств человека при использовании очков виртуальной реальности. – Прим. перев.

людьми и препятствиями на пути (Berg et al., 2015). В данной работе обнаружение аномалий на пути осуществляется с помощью фильтра на основе MOSSE. Кроме того, системы безопасности автомобилей, которые, например, обнаруживают пешеходов с помощью ИК-камер (Källhammer et al., 2007) и RGB-камер, выигрывают от прогресса в области комбинированного RGB-и ИК-отслеживания (Kristan et al., 2019b).

Методика DCF подходит не только для отслеживания и обнаружения аномалий, но и может быть дополнительно обобщена в направлении масштабно-пространственного отслеживания. Например, визуальная навигация судна в прибрежных районах может обеспечить точность GPS (Grelsson et al., 2020). Здесь эффективность реализации DCF-преобразования Фурье используется для сопоставления в режиме реального времени сегмента линии горизонта с огромным количеством модельных горизонтов, выбранных из цифровой модели рельефа.

С переходом методов VOS и VIS на обработку в реальном времени возникают новые области применения и новые функции в существующих областях. Например, системы дополненной реальности могут использовать совмещение сегментированных объектов для дальнейшего улучшения точности и задержки. Автономные автомобили и передовые системы помощи водителю уже сейчас используют семантическую сегментацию по кадрам, но быстрое действие может еще больше возрасти благодаря мощным алгоритмам VIS. Навигация судов больше не нуждается в отдельном извлечении линии горизонта, а может интегрировать анализ горизонта в сопоставление изображений.

Очевидно, что потенциальное неправомерное использование и этически неоднозначные способы применения также выигрывают от развития современных методов компьютерного зрения. Например, массовая слежка за людьми со стороны органов власти или компаний уже стала заурядным явлением. Точно так же методы VOS можно использовать для фальсификации видео или создания дипфейков. Наконец, существуют различные военные применения, в частности с использованием ИК-изображений. Открытость исследований и доступность реализаций с открытым исходным кодом являются ключом к научному прогрессу, но, к сожалению, также и к неправомерному использованию его плодов.

БЛАГОДАРНОСТИ

Большая часть этой главы посвящена методам и результатам, полученным в течение последних семи лет в Лаборатории компьютерного зрения в Линчепинге в сотрудничестве с Фахадом Ханом и несколькими аспирантами и магистрантами. В первую очередь следует упомянуть Мартина Данельяна, Густава Хегера, Андреаса Робинсона, Йоакима Йонандера, Феликса Яремолавина, Гутама Бхата, Аманду Берг и Сюзанну Глад.

Исследовательская работа, описанная в этой главе, была частично поддержана KAW через Wallenberg AI, программой Autonomous Systems and Software

Program WASP, Шведским исследовательским советом через проекты EMC2, NCNN и ELLIIT, SSF через проекты CUAS и Symbicloud, а также LiU через CE-NIIT.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Baker S., Matthews I.*, 2004. Lucas-Kanade 20 years on: a unifying framework. *International Journal of Computer Vision* 56 (3), 221–255.
- Berg A., Öffäll K., Ahlberg J., Felsberg M.*, 2015. Detecting rails and obstacles using a train-mounted thermal camera. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. In: LNCS, vol. 9127, pp. 492–503.
- Berg A., Johnander J., Durand De Gevigney F., Ahlberg J., Felberg M.*, 2019. Semi-automatic annotation of objects in visual-thermal video. In: *Proceedings – 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pp. 2242–2251.
- Bhat G., Johnander J., Danelljan M., Khan F. S., Felsberg M.*, 2018. Unveiling the power of deep tracking. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. In: LNCS, vol. 11206, pp. 493–509.
- Bhat G., Danelljan M., Van Gool L., Timofte R.*, 2019. Learning discriminative model prediction for tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 6181–6190.
- Bhat G., Lawin F. J., Danelljan M., Robinson A., Felsberg M., Van Gool L., Timofte R.*, 2020. Learning what to learn for video object segmentation. In: *Vedaldi A., Bischof H., Brox T., Frahm J.-M. (Eds.), Computer Vision – ECCV 2020*. Springer International Publishing, Cham, pp. 777–794.
- Bolme D. S., Beveridge J. R., Draper B. A., Lui Y. M.*, 2010. Visual object tracking using adaptive correlation filters. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550.
- Bracewell R. N.*, 1995. *Two-Dimensional Imaging*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs.
- Chandaria J., Thomas G., Bartczak B., Koeser K., Koch R., Becker M., Bleser G., Stricker D., Wohlleber C., Felsberg M., Gustafsson F., Hol J. D., Schön T. B., Skoglund J., Slycke P. J., Smeitz S.*, 2007. Realtime camera tracking in the MATRIS project. *SMPTE Motion Imaging Journal* 116 (7–8), 266–271.
- Chatfield K., Simonyan K., Vedaldi A., Zisserman A.*, 2014. Return of the devil in the details: delving deep into convolutional nets. In: *British Machine Vision Conference*.
- Chen Q.-S., DeFrise M., Deconinck F.*, 1994. Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 1156–1168.
- Chéron G., Laptev I., Schmid C.*, 2015. P-CNN: pose-based CNN features for action recognition. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3218–3226.

- Dalal N., Triggs B.*, 2005. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 886–893.
- Danelljan M., Häger G., Khan F. S., Felsberg M.*, 2014a. Accurate scale estimation for robust visual tracking. In: BMVC 2014 – Proceedings of the British Machine Vision Conference 2014.
- Danelljan M., Khan F. S., Felsberg M., Van De Weijer J.*, 2014b. Adaptive color attributes for real-time visual tracking. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1090–1097.
- Danelljan M., Hager G., Khan F. S., Felsberg M.*, 2015. Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2015 Inter, pp. 4310–4318.
- Danelljan M., Häger G., Khan F. S., Felsberg M.*, 2016a. Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem, 1430–1438.
- Danelljan M., Hager G., Khan F. S., Felsberg M.*, 2016b. Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2016-Febru, pp. 621–629.
- Danelljan M., Robinson A., Khan F., Felsberg M.*, 2016. Beyond correlation filters: Learning continuous convolution operators for visual tracking. LNCS, vol. 9909.
- Danelljan M., Hager G., Khan F. S., Felsberg M.*, 2017. Discriminative scale space tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (8), 1561–1575.
- Danelljan M., Bhat G., Gladh S., Khan F. S., Felsberg M.*, 2019a. Deep motion and appearance cues for visual tracking. Pattern Recognition Letters 124, 74–81.
- Danelljan M., Bhat G., Khan F. S., Felsberg M.*, 2019b. Atom: accurate tracking by overlap maximization. Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2019-June, 4655–4664.
- Dendorfer P., Osep A., Milan A., Schindler K., Cremers D., Reid I., Roth S., Leal-Taixé L.*, 2020. MOTChallenge: a benchmark for single-camera multiple target tracking. International Journal of Computer Vision.
- Everingham M., Van Gool L., Williams C. K., Winn J., Zisserman A.*, 2010. The Pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88 (2), 303–338.
- Felsberg M.*, 1998. Signal Processing Using Frequency Domain Methods in Clifford Algebra. Diploma thesis Institute of Computer Science and Applied Mathematics. Christian-Albrechts-University of Kiel.
- Felsberg M.*, 2009. Spatio-featural scale-space. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). In: LNCS, vol. 5567, pp. 808–819.
- Felsberg M.*, 2013. Enhanced distribution field tracking using channel representations. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 121–128.
- Felsberg M.*, 2018. Probabilistic and Biologically Inspired Feature Representations. Morgan & Claypool Publishers.

- Felsberg M., Duits R., Florack L.*, 2005. The monogenic scale space on a rectangular domain and its features. *International Journal of Computer Vision* 64 (2–3), 187–201.
- Galoogahi H. K., Sim T., Lucey S.*, 2015. Correlation filters with limited boundaries. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4630–4638.
- Garon M., Lalonde J.-F.*, 2017. Deep 6-DOF tracking. *IEEE Transactions on Visualization and Computer Graphics* 23, 2410–2418.
- Girshick R., Donahue J., Darrell T., Malik J.*, 2016. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 142–158.
- Grelsson B., Robinson A., Felsberg M., Khan F. S.*, 2020. GPS-level accurate camera localization with HorizonNet. *Journal of Field Robotics* 37 (6), 951–971.
- Häger G., Bhat G., Danelljan M., Khan F. S., Felsberg M., Rudl P., Doherty P.*, 2016. Combining visual tracking and person detection for long term tracking on a UAV. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. In: LNCS, vol. 10072, pp. 557–568.
- Häger G., Felsberg M., Khan F.*, 2018. Countering bias in tracking evaluations. In: *VISIGRAPP 2018 – Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, pp. 581–587.
- He K., Zhang X., Ren S., Sun J.*, 2016. Deep residual learning for image recognition. In: *CVPR*.
- Henriques J. F., Caseiro R., Martins P., Batista J.*, 2012. Exploiting the circulant structure of tracking-by-detection with kernels. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. In: LNCS, vol. 7575, pp. 702–715.
- Horner J. L., Gianino P. D.*, 1984. Phase-only matched filtering. *Applied Optics* 23 (6), 812–816.
- Jaccard P.*, 1912. The distribution of the flora in the Alpine zone. *New Phytologist* 11 (2), 37–50.
- Johnander J., Danelljan M., Brissman E., Khan F. S., Felsberg M.*, 2019. A generative appearance model for endtoend video object segmentation. *Proceedings - IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June, 8945–8954.
- Källhammer J. E., Eriksson D., Granlund G., Felsberg M., Moe A., Johansson B., Wiklund J., Forssén P. E.*, 2007. Near zone pedestrian detection using a low-resolution FIR sensor. In: *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 339–345.
- Khan F. S., Anwer R. M., van de Weijer J., Bagdanov A., Vanrell M., Lopez A. M.*, 2012. Color attributes for object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Koenderink J. J.*, 1984. The structure of images. *Biological Cybernetics* 50, 363–370.
- Kristan M., Pflugfelder R., Leonardis A., Matas J., Porikli F., Čehovin L., Nebehay G., Fernandez G., Vojří T., Gatt A., Khajenezhad A., Salahledin A., Soltani-Farani A., Zaregade A., Petrosino A., Milton A., Bozorgtabar B., Li B., Chan C. S., Heng C.*

- Ward D., Kearney D., Monekosso D., Karaimer H. C., Rabiee H. R., Zhu J., Gao J., Xiao J., Zhang J., Xing J., Huang K., Lebeda K., Cao L., Maresca M. E., Lim M. K., El-Helw M., Felsberg M., Remagnino P., Bowden R., Goecke R., Stolkin R., Lim S. Y. Y., Maher S., Poullot S., Wong S., Satoh S., Chen W., Hu W., Zhang X., Li Y., Niu Z., 2013. The visual object tracking VOT2013 challenge results. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 98–111.
- Kristan M., Pflugfelder R., Leonardis A., Matas J., Čehovin L., Nebehay G., Vojíř T., Fernández G., Lukežič A., Dimitriev A., Petrosino A., Saffari A., Li B., Han B., Heng C. K., Garcia C., Pangeršič D., Häger G., Khan F. S., Oven F., Possegger H., Bischof H., Nam H., Zhu J., Li J. J., Choi J. Y., Choi J. W., Henriques J. F., van de Weijer J., Batista J., Lebeda K., Öfjäll K., Yi K. M., Qin L., Wen L., Maresca M. E., Danelljan M., Felsberg M., Cheng M. M., Torr P., Huang Q., Bowden R., Hare S., Lim S. Y. Y., Hong S., Liao S., Hadfield S., Li S. Z., Duffner S., Golodetz S., Mauthner T., Vineet V., Lin W., Li Y., Qi Y., Lei Z., Niu Z. H., 2015a. The visual object tracking VOT2014 challenge results, vol. 8926.
- Kristan M., Matas J., Leonardis A., Felsberg M., Čehovin L., Fernández G., Vojíř T., Häger G., Nebehay G., Pflugfelder R., Gupta A., Bibi A., Lukežič A., Garcia-Martin A., Saffari A., Petrosino A., Montero A., Varfolomieiev A., Baskurt A., Zhao B., Ghanem B., Martinez B., Lee B., Han B., Wang C., Garcia C., Zhang C., Schmid C., Tao D., Kim D., Huang D., Prokhorov D., Du D., Yeung D.-Y., Ribeiro E., Khan F., Porikli F., Bunyak F., Zhu G., Seetharaman G., Kieritz H., Yau H., Li H., Qi H., Bischof H., Possegger H., Lee H., Nam H., Bogun I., Jeong J.-C., Cho J.-I., Lee J. Y., Zhu J., Shi J., Li J., Jia J., Feng J., Gao J., Choi J., Kim J.-W., Lang J., Martinez J., Choi J., Xing J., Xue K., Palaniappan K., Lebeda K., Alahari K., Gao K., Yun K., Wong K., Luo L., Ma L., Ke L., Wen L., Bertinetto L., Pootschi M., Maresca M., Danelljan M., Wen M., Zhang M., Arens M., Valstar M., Tang M., Chang M.-C., Khan M., Fan N., Wang N., Miksik O., Torr P., Wang Q., Martin-Nieto R., Pelapur R., Bowden R., Laganière R., Moujtahid S., Hare S., Hadfield S., Lyu S., Li S., Zhu S.-C., Becker S., Duffner S., Hicks S., Golodetz S., Choi S., Wu T., Mauthner T., Pridmore T., Hu W., Hübner W., Wang X., Li X., Shi X., Zhao X., Mei X., Shizeng Y., Hua Y., Li Y., Lu Y., Li Y., Chen Z., Huang Z., Chen Z., Zhang Z., He Z., Hong Z., 2015b. The visual object tracking VOT2015 challenge results. In: *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015-Febru.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Čehovin L., Vojíř T., Häger G., Lukežič A., Fernández G., Gupta A., Petrosino A., Memarmoghdam A., Martin A. G., Montero A. S., Vedaldi A., Robinson A., Ma A. J., Varfolomieiev A., Alatan A., Erdem A., Ghanem B., Liu B., Han B., Martinez B., Chang C. M., Xu C., Sun C., Kim D., Chen D., Du D., Mishra D., Yeung D. Y., Gundogdu E., Erdem E., Khan F., Porikli F., Zhao F., Bunyak F., Battistone F., Zhu G., Roffo G., Sai Subrahmanyam G. R., Bastos G., Seetharaman G., Medeiros H., Li H., Qi H., Bischof H., Possegger H., Lu H., Lee H., Nam H., Chang H. J., Drummond I., Valmadre J., Jeong J. C., Cho J. I., Lee J. Y., Zhu J., Feng J., Gao J., Choi J. Y., Xiao J., Kim J. W., Jeong J., Henriques J. F., Lang J., Choi J., Martinez J. M., Xing J., Gao J., Palaniappan K., Lebeda K., Gao K., Mikolajczyk K., Qin L., Wang L., Wen L., Bertinetto L., Rapuru M. K., Poostchi M., Maresca M., Danelljan M., Mueller M., Zhang M., Arens M., Valstar M., Tang M., Baek M., Khan M. H., Wang N., Fan N., Al-Shakrji N., Miksik O., Akin O., Moallem P., Senna P., Torr P. H., Yuen P. C., Huang Q.,

- Nieto R. M., Pelapur R., Bowden R., Laganière R., Stolkin R., Walsh R., Krah S. B., Li S., Zhang S., Yao S., Hadfield S., Melzi S., Lyu S., Li S., Becker S., Golodetz S., Kakanuru S., Choi S., Hu T., Mauthner T., Zhang T., Pridmore T., Santopietro V., Hu W., Li W., Hübner W., Lan X., Wang X., Li X., Li Y., Demiris Y., Wang Y., Qi Y., Yuan Z., Cai Z., Xu Z., He Z., Chi Z., 2016. The visual object tracking VOT2016 challenge results. LNCS, vol. 9914. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Chovín, L., 2016b. A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Zajc L., Vojř T., Häger G., Lukežič A., Eldesokey A., Fernández G., García-Martín Á., Muhic A., Petrosino A., Memarmoghadam A., Vedaldi A., Manzanera A., Tran A., Alatan A., Mocanu B., Chen B., Huang C., Xu C., Sun C., Du D., Zhang D., Du D., Mishra D., Gundogdu E., Velasco-Salido E., Khan F., Battistone F., Subrahmanyam G., Bhat G., Huang G., Bastos G., Seetharaman G., Zhang H., Li H., Lu H., Drummond I., Valmadre J., Jeong J.-C., Cho J.-I., Lee J.-Y., Noskova J., Zhu J., Gao J., Liu J., Kim J.-W., Henriques J., Martínez J., Zhuang J., Xing J., Gao J., Chen K., Palaniappan K., Lebeda K., Gao K., Kitani K., Zhang L., Wang L., Yang L., Wen L., Bertinetto L., Poostchi M., Danelljan M., Mueller M., Zhang M., Yang M.-H., Xie N., Wang N., Miksik O., Moallem P., Pallavi Venugopal M., Senna P., Torr P., Wang Q., Yu Q., Huang Q., Martín-Nieto R., Bowden R., Liu R. Tapu, R. Hadfield S., Lyu S., Golodetz S., Choi S., Zhang T., Zaharia T., Santopietro V., Zou W., Hu W., Tao W., Li W., Zhou W., Yu X., Bian X., Li Y., Xing Y., Fan Y., Zhu Z., Zhang Z., He Z., 2017. The visual object tracking VOT2017 challenge results. In: Proceedings – 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017, vol. 2018-Janua.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Zajc L., Vojř T., Bhat G., Lukežič A., Eldesokey A., Fernández G., García-Martín Á., Iglesias-Arias Á., Alatan A., González-García A., Petrosino A., Memarmoghadam A., Vedaldi A., Muhic A., He A., Smeulders A., Perera A., Li B., Chen B., Kim C., Xu C., Xiong C., Tian C., Luo C., Sun C., Hao C., Kim D., Mishra D., Chen D., Wang D., Wee D., Gavves E., Gundogdu E., Velasco-Salido E., Khan F., Yang F., Zhao F., Li F., Battistone F., De Ath G., Subrahmanyam G., Bastos G., Ling H., Galoogahi H., Lee H., Li H., Zhao H., Fan H., Zhang H., Possegger H., Li H., Lu H., Zhi H., Li H., Lee H., Chang H., Drummond I., Valmadre J., Martin J., Chahl J., Choi J., Li J., Wang J., Qi J., Sung J., Johnander J., Henriques J., Choi J., van de Weijer J., Herranz J., Martínez J., Kittler J., Zhuang J., Gao J., Grm K., Zhang L., Wang L., Yang L., Rout L., Si L., Bertinetto L., Chu L., Che M., Maresca M., Danelljan M., Yang M.-H., Abdelpakey M., Shehata M., Kang M., Lee N., Wang N., Miksik O., Moallem P., Vicente-Moñivar P., Senna P., Li P., Torr P., Raju P., Ruihe Q., Wang Q., Zhou Q., Guo Q., Martín-Nieto R., Gorthi R., Tao R., Bowden R., Everson R., Wang R., Yun S., Choi S., Vivas S., Bai S., Huang S., Wu S., Hadfield S., Wang S., Golodetz S., Ming T., Xu T., Zhang T., Fischer T., Santopietro V., Štruc V., Wei W., Zuo W., Feng W., Wu W., Zou W., Hu W., Zhou W., Zeng W., Zhang X., Wu X., Wu X.-J., Tian X., Li Y., Lu Y., Law Y., Wu Y., Demiris Y., Yang Y., Jiao Y., Li Y., Zhang Y., Sun Y., Zhang Z., Zhu Z., Feng Z.-H., Wang Z., He Z., 2019. The sixth visual object tracking VOT2018 challenge results. LNCS, vol. 11129.

- Kristan M., Matas J., Leonardis A., Felsberg M., Pflugfelder R., Kämäräinen J.-K., Zajc L., Drbohlav O., Lukežić A., Berg A., Eldesokey A., Kapyla J., Fernández G., Gonzalez-Garcia A., Memarmoghadam A., Lu A., He A., Varfolomieiev A., Chan A., Tripathi A., Smeulders A., Pedasingu B., Chen B., Zhang B., Baoyuanwu B., Li B., He B., Yan B., Bai B., Li B., Li B., Kim B., Ma C., Fang C., Qian C., Chen C., Li C., Zhang C., Tsai C.-Y., Luo C., Micheloni C., Zhang C., Tao D., Gupta D., Song D., Wang D., Gavves E., Yi E., Khan F., Zhang F., Wang F., Zhao F., De Ath G., Bhat G., Chen G., Wang G., Li G., Cevikalp H., Du H., Zhao H., Saribas H., Jung H., Bai H., Yu H., Peng H., Lu H., Li H., Li J., Li J., Fu J., Chen J., Gao J., Zhao J., Tang J., Li J., Wu J., Liu J., Wang J., Qi J., Zhang J., Tsotsos J., Lee J., Van De Weijer J., Kittler J., Ha Lee J., Zhuang J., Zhang K., Wang K., Dai K., Chen L., Liu L., Guo L., Zhang L., Wang L., Wang L., Zhang L., Wang L., Zhou L., Zheng L., Rout L., Van Gool L., Bertinetto L., Danelljan M., Dunnhofer M., Ni M., Kim M., Tang M., Yang M.-H., Paluru N., Martinel N., Xu P., Zhang P., Zheng P., Zhang P., Torr P., Wang Q., Guo Q., Timofte R., Gorthi R., Everson R., Han R., Zhang R., You S., Zhao S.-C., Zhao S., Li S., Li S., Ge S., Bai S., Guan S., Xing T., Xu T., Yang T., Zhang T., Vojtík T., Feng W., Hu W., Wang W., Tang W., Zeng W., Liu W., Chen X., Qiu X., Bai X., Wu X.-J., Yang X., Chen X., Li X., Sun X., Chen X., Tian X., Tang X., Zhu X. F., Huang Y., Chen Y., Lian Y., Gu Y., Liu Y., Chen Y., Zhang Y., Xu Y., Wang Y., Li Y., Zhou Y., Dong Y., Xu Y., Zhang Y., Li Y., Luo Z., Zhang Z., Feng Z.-H., He Z., Song Z., Chen Z., Zhang Z., Wu Z., Xiong Z., Huang Z., Teng Z., Ni Z., 2019b. The seventh visual object tracking VOT2019 challenge results. In: *Proceedings – 2019 International Conference on Computer Vision Workshop, ICCVW 2019*.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Kamarainen J.-K., Zajc L. C., Danelljan M., Lukežić A., Drbohlav O., He L., Zhang Y., Yan S., Yang J., Fernandez G., et al., 2020. The eighth visual object tracking VOT2020 challenge results.
- Lowe D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60 (2), 91–110.
- Lucas B. D., Kanade T., 1981. An iterative image registration technique with an application to stereo vision. In: *Proceedings of International Joint Conference on Artificial Intelligence*.
- Lukežić A., Zajc L. C., Kristan M., 2018. Fast Spatially Regularized Correlation Filter Tracker.
- Matthews L., Ishikawa T., Baker S., 2004. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (6), 810–815.
- Neumann U., Park J., 1998. Extendible object-centric tracking for augmented reality. In: *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium* (Cat. No. 98CB36180), pp. 148–155.
- Neumann U., You S., Cho Y., Lee J., Park J., 1999. Augmented reality tracking in natural environments. In: *International Symposium on Mixed Realities*.
- Perazzi F., Pont-Tuset J., McWilliams B., Van Gool L., Gross M., Sorkine-Hornung A., 2016. A benchmark dataset and evaluation methodology for video object segmentation. In: *Computer Vision and Pattern Recognition*.
- Perazzi F., Khoreva A., Benenson R., Schiele B., Sorkine-Hornung A., 2017. Learning video object segmentation from static images. In: *Proceedings – 30th*

- IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 3491–3500.
- Pietikäinen M., Zhao G.*, 2015. Two decades of local binary patterns: a survey. *Advances in Independent Component Analysis and Learning Machines* abs/1612.0, 175–210.
- Reiter R.*, 1978. *On ClosedWorld Data Bases*. Springer US, Boston, MA, pp. 55–76.
- Ripley D. L., Politzer T.*, 2010. Vision disturbance after TBI. *NeuroRehabilitation* 27, 215–216.
- Robinson A., Lawin F. J., Danelljan M., Khan F. S., Felsberg M.*, 2020. Learning fast and robust target models for video object segmentation. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7404–7413.
- Shi Jianbo Tomasi*, 1994. Good features to track. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.
- Simonyan K., Zisserman A.*, 2015. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations*.
- Skoglund J., Felsberg M.*, 2006. Evaluation of subpixel tracking algorithms. *LNCS*, vol. 4292.
- Skoglund J., Felsberg M.*, 2007. Covariance estimation for SAD block matching. *LNCS*, vol. 4522.
- Stauffer C., Grimson W. E. L.*, 2000. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8), 747–757.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.*, 2015. Going deeper with convolutions. In: *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 1–9.
- Van De Weijer J., Schmid C., Verbeek J., Larlus D.*, 2009. Learning color names for real-world applications. *IEEE Transactions on Image Processing* 18 (7), 1512–1523.
- Vojír T., Matas J.*, 2017. Pixel-wise object segmentations for the VOT 2016 dataset. In: *Research Reports of CMP*, no. 1.
- Wang B., Qi Z., Chen S.*, 2016. Motion-based feature selection and adaptive template update strategy for robust visual tracking. In: *Proceedings – 2016 3rd International Conference on Information Science and Control Engineering, ICISCE 2016*, vol. 1, pp. 462–467.
- Wu Y., Lim J., Yang M. H.*, 2013. Online object tracking: a benchmark. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418.
- Yang L., Fan Y., Xu N.*, 2019. Video instance segmentation. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5187–5196.
- Yao R., Lin G., Xia S., Zhao J., Zhou Y.*, 2019. Video object segmentation and tracking: a survey. *arXiv* 1 (1).
- Zhu Z., Wang Q., Li B., Wu W.*, 2018. Distractor-aware Siamese networks for visual object tracking. In: *Eccv 2018*, pp. 1–17. *arXiv:1808.06048v1 [cs.CV]*.

ОБ АВТОРЕ ГЛАВЫ

Майкл Фельсберг – профессор компьютерного зрения на факультете электроники Университета Линчепинга, Швеция. Он также является почетным профессором Инженерной школы Университета Квазулу-Натал в Дурбане, Южная Африка. Он имеет степень доктора философии Кильского университета, Германия (2002 г.), и степень доцента Университета Линчепинга (2005 г.). Он получил различные награды, в том числе премию Olympus (2005 г.) от DAGM и награды за лучшую работу от ICPR (2016 г.) и VISAPP (2021 г.). Его индекс Хирша в Google Scholar равен 44. Его исследовательские интересы включают, помимо визуального отслеживания объектов, сегментацию видеообъектов и экземпляров, обработку облака точек и эффективные методы машинного обучения.

Глава 10

.....

Длительное отслеживание объекта на основе глубокого обучения

Авторы главы:
Эфстратиос Гаввес, Дипак Гупта,
Институт информатики Амстердамского университета,
Амстердам, Нидерланды

Краткое содержание главы:

- определение задачи отслеживания видеообъектов с точки зрения машинного обучения и оптимизации;
- краткое изложение проблем визуального восприятия, обучения и технической реализации отслеживания объектов;
- обзор современных средств краткосрочного визуального отслеживания объектов, включая их ограничения, когда речь идет о более длинных и сложных пространственно-временных видеопотоках;
- введение в долгосрочное визуальное отслеживание объектов и проблему негативных последствий, связанных с распадом модели, появлением и исчезновением цели;
- описание глубоких сиамских трекеров, которые представляют современное состояние визуального отслеживания объектов, особенно в длительном отслеживании видеопотоков;
- обзор инвариантности и эквивариантности представлений с обсуждением того, как они соотносятся со средствами глубокого визуального отслеживания объектов.

Я посвящаю эту работу моей жене Катерине, моему сыну Ясонасу, моей матери Антонию, моему брату Манолису и, конечно же, моему отцу Гавриилу, который всегда будет в наших сердцах и мыслях.

10.1. ВВЕДЕНИЕ

Глубокое обучение за последнее десятилетие стало причиной радикальных перемен в области компьютерного зрения, от распознавания объектов, семантической сегментации и реконструкции 3D-поверхности до смыслового понимания видео. Будучи одной из старейших задач компьютерного зрения, отслеживание видеообъектов также достигло значительного прогресса в эпоху глубокого обучения, хотя и с задержкой по причинам, которые мы опишем позже в этой главе. Глубокое обучение не только значительно повышает количественные показатели отслеживания объектов, но и улучшает точность прохождения бенчмарков. Важно отметить, что прогресс был именно качественным, что позволило перейти в отслеживании объектов от видеороликов длиной в несколько секунд, более известных как *краткосрочное отслеживание* (раздел 10.2), к видеороликам, охватывающим несколько десятков минут и относящимся к *длительному отслеживанию* (раздел 10.3).

Визуальное отслеживание объекта в его наиболее общей форме можно описать как обучение модели f_ϕ с параметрами ϕ , которая пытается предсказать будущее местоположение цели отслеживания с учетом ее известного начального положения, то есть

$$f_\phi : \mathcal{Y}_1 \times \mathcal{X}_1 \times \mathcal{X}_t \rightarrow \mathcal{Y}_t. \quad (10.1)$$

\mathcal{Y}_t обозначает пространство всех возможных прогнозов, будь то в форме ограничивающей рамки или маски сегментации. \mathcal{X}_t обозначает пространство всех возможных входных кадров в момент t . Единственная исходная информация, известная модели, – это расположение объекта на первом кадре.

Часто модели трекеров используют свои собственные промежуточные прогнозы y_i , $i < t$, для обновления параметров модели ϕ . В этом случае параметры модели также изменяются во времени и лучше представлены с помощью ϕ_t . Однако, как мы поясним позже, важно помнить, что даже если промежуточные прогнозы y_i используются в качестве целевых переменных для обновления моделей с помощью алгоритмов обучения с учителем, эти переменные не совпадают с истинным местоположением объекта: $y_i \neq y_i^*$.

При оптимизации любого средства визуального отслеживания объектов наиболее часто исходят из критерия *минимизации эмпирического риска*. В частности, функция потерь определяется соответствующей моделью выбора

$$\mathcal{L} = \mathcal{L}(\phi_t; \mathbf{x}_{1:t}, y_1^*, y_{1:t-1}), \quad (10.2)$$

минимизированной с помощью (стохастического) градиентного спуска или его вариантов,

$$\operatorname{argmin}_{\phi_t} \mathcal{L} \Rightarrow \phi_{t+1} = \phi_t - \epsilon \frac{\partial \mathcal{L}}{\partial \phi}. \quad (10.3)$$

При необходимости функция потерь дополняется функцией регуляризации $\Omega(\phi)$, которая штрафует веса в соответствии с заранее определенными

принципами архитектуры модели, например штрафую переобучение или неразрезанные решения.

Хотя фундаментальные принципы работы трекеров неизменны, в последнее время трекеры делятся на краткосрочные и долгосрочные. Согласно каноническому определению в (Kristan et al., 2016), *краткосрочное отслеживание не требует применения методов повторного обнаружения цели*. То есть предполагается, что целевой объект не исчезает из кадра и не появляется снова. Напротив, при длительном визуальном сопровождении объекта целевой объект может исчезнуть и снова появиться в кадре в любой момент. Возможно, что более важно, длительное отслеживание связано со сложными и длинными видео, где объект может претерпевать серьезные искажения внешнего вида. Это могут быть изменения самого объекта, когда ракурс обзора целевого объекта постоянно и сильно меняется, приводя к изменению внешнего вида объекта в кадре, притом что трекер имеет в своем распоряжении только внешний вид цели в начальный момент времени. Эти искажения также могут быть вызваны изменениями в окружающей среде, например в способе освещения сцены или из-за окклюзии (частичного заслонения) другими объектами в сцене. Аналогичные помехи, естественно, проявляются и при краткосрочном отслеживании, и на первый взгляд их устранение выглядит как прямое расширение методики на более длинные видео. Однако, как мы покажем далее, эти возмущения имеют косвенные последствия, которые уникальны для длительного отслеживания просто из-за большей продолжительности, часто являются накопительными и могут даже оказаться катастрофическими для рассматриваемых моделей.

10.1.1. Трудности отслеживания видеообъектов

На первый взгляд визуальное отслеживание объекта кажется довольно простой задачей. В конце концов, люди с легкостью непрерывно следят за объектом, независимо от его типовой принадлежности, внешнего вида и ситуаций, в которых он появляется. На самом деле отслеживание – одна из самых сложных задач компьютерного зрения и прикладного машинного обучения. Во многом это связано с тем, что отслеживание представляет собой задачу с крайне размытыми условиями: в общем случае почти не существует ограничений на типы объектов или вариации сцен на протяжении видео.

Далее мы определяем и описываем три типа проблем визуального отслеживания объектов: *видовые*, *обучающие* и *технические*. Важно подчеркнуть, что эти проблемы редко – если вообще когда-либо – встречаются по отдельности.

10.1.1.1. Видовые проблемы отслеживания

Мы выделяем два типа видовых проблем отслеживания: *внутренние*, вызванные изменениями самого целевого объекта, и *внешние*, вызванные изменениями в среде отслеживания.

Начнем с внутренних видовых проблем. Обширные исследования (Smeulders et al., 2014; Wu et al., 2015) позволили составить перечень изменений,

характерных для объекта: *изменение масштаба, деформация, вращение в плоскости кадра и вращение вне плоскости*. Масштаб меняется либо при изменении размера объекта, либо при изменении расстояния между объектом и камерой. Деформация обычно наблюдается, когда целевой объект не является жестким и со временем меняет форму. Вращение в плоскости кадра – это повороты либо целевого объекта, либо камеры в плоскости, перпендикулярной оси между камерой и целевым объектом. Примером вращения в плоскости кадра является запись пешеходов с высоты птичьего полета, например с дрона, когда пешеходы меняют направление. Напротив, вращения вне плоскости – это все остальные вращения, происходящие в трехмерном пространстве, например при отслеживании танцующего человека.

К изменениям, вызванным окружающей средой, мы относим *изменение освещения, окклюзию, исчезновение и повторное появление цели и изменение фона*. Мы говорим об изменении освещения, когда освещенность цели меняется на протяжении всей последовательности видеок кадров, что может привести к существенному изменению внешнего вида объекта и, возможно, даже его видимого цвета. Например, внешний вид автомобиля, ехавшего по дороге, а затем нырнувшего в туннель, существенно изменится из-за уменьшения контраста видеок кадров и цвета фонарей освещения. Частичная или полная окклюзия вызвана другими объектами в сцене, которые могут быть движущимися или статичными. Интересно, что окклюзия – это особая разновидность изменений, потому что она связана с *отсутствием* информации и не может быть легко изучена в явном виде. Подобно окклюзии, исчезновение и повторное появление цели происходят, когда цель выходит из кадра и снова становится видимой в случайной точке в будущем, возможно даже с разными точками выхода и входа, например когда автомобиль покидает кадр с левой стороны и снова появляется справа. Хотя алгоритмы краткосрочного отслеживания часто предполагают отсутствие исчезновения и повторного появления цели (Kristan et al., 2016), при длительном отслеживании такое предположение сделать нельзя. Наконец, отдельной проблемой является изменение фона, потому что со статистической точки зрения оно несет в себе риск обнаружения случайных визуальных паттернов, которые можно временно спутать с паттернами целевого объекта. Это может быть особенно проблематично в случае моделей, которые рассчитаны на интенсивное обновление.

10.1.1.2. Проблемы машинного обучения при отслеживании

В общем случае визуальное отслеживание объектов не накладывает никаких ограничений на типы или внешний вид объектов. Одно видео может быть о машине, путешествующей по шоссе, а другое – о поведении осьминога, который деформируется любым мыслимым образом. Если на видео представлен процесс медицинского обследования, отслеживаемый объект может даже не иметь обычного внешнего вида, формы или характерных движений, присущих обычным объектам. В принципе, модель трекера должна иметь возможность отслеживать все эти объекты на протяжении всей последовательности без каких-либо дополнительных указаний, кроме определения

цели пользователем в первом кадре. Это подводит нас к следующей проблеме обучения: в отличие от стандартных задач машинного обучения, таких как классификация, обнаружение или сегментация объектов, при визуальном отслеживании объектов моделируемые целевые объекты являются произвольными и не могут быть определены заранее. То есть при визуальном отслеживании объекта у нас не может быть обучающих образцов отслеживаемого объекта, поскольку тогда это будет уже не отслеживание, а *обнаружение* объекта. В этом заключается принципиальное отличие от большинства задач машинного обучения, где обычно есть обучающие и тестовые наборы, содержащие разные образцы данных из одних и тех же категорий объектов. С другой стороны, визуальное отслеживание объектов сродни популярным в последнее время парадигмам обучения за несколько шагов и за один проход (Bertinetto et al., 2016), которые являются одними из самых сложных вариантов машинного обучения.

Визуальное отслеживание объектов в основном моделируется с помощью методологий обучения с учителем. Однако из-за принципиального отсутствия достаточного количества положительных образцов для обучения надежных обобщающих моделей обычные трекеры полагаются на извлечение данных из первого кадра и обновление модели в последующих кадрах. Чтобы модель могла дообучаться на собственных прогнозах, приходится считать их псевдоположительными, что постепенно приводит к увеличению смещения модели трекера. Мы обсудим *устаревание модели* (model decay) более подробно позже в этой главе, так как это, пожалуй, самая сложная проблема, когда речь идет о долгосрочном отслеживании объекта.

Визуальное отслеживание – это не только локализация целевого объекта, но и отделение его от фона. Однако сцены, в которых появляются объекты, могут резко меняться на протяжении видео. Ситуацию значительно усложняет наличие нескольких объектов с похожим или даже идентичным внешним видом, например при отслеживании конкретного спортсмена во время футбольного матча или человека в марширующем оркестре. Во время отслеживания фон может сильно отличаться от одной последовательности к другой и еще сильнее меняться с течением времени. Это подводит нас ко второй проблеме обучения: при визуальном отслеживании объектов модель должна научиться моделировать положительные образцы, то есть целевой объект, и игнорировать отрицательные образцы, то есть фон (Bhat et al., 2019). Фон может существенно отличаться от того, что наблюдалось в предыдущих видео. Фактически фон может постоянно меняться с течением времени в пределах одного видео и сильно отвлекать модель или даже сбивать ее с толку. В этом смысле визуальное отслеживание объектов также похоже на обнаружение аномалий, когда модель машинного обучения подвергается воздействию преимущественно или исключительно положительных примеров и должна научиться отличать их от всех других возможных отрицательных примеров.

Наконец, критической проблемой обучения является настройка гиперпараметров. В отличие от других задач компьютерного зрения и машинного обучения, отслеживание визуального объекта часто относится к сценариям, в которых входные данные нестационарны и недоступны заранее. В резуль-

тате гиперпараметры, оптимальные для одного видео, могут оказаться совершенно неоптимальными для других. На практике часто случается так, что при небольших изменениях в модели трекера точность отслеживания улучшается в одних видео, но падает в других. Выбор подходящего типа модели и настройка ее гиперпараметров часто имеют решающее значение для робастности и обобщающей способности.

10.1.1.3. Технические проблемы при отслеживании

Помимо проблем, связанных с внешним видом цели и фона, а также с моделями машинного обучения, при визуальном отслеживании объектов существуют технические проблемы, связанные со способом записи видеоряда или устройством модели трекера.

Техническая проблема, которая часто приводит к сбою трекера, – это быстрое движение, когда скорость целевого объекта выше, чем частота кадров. Быстрое движение чаще всего доставляет проблемы моделям трекеров, которые полагаются на ограничение радиуса поиска вокруг своих предыдущих прогнозов. Как правило, радиус поиска берется достаточно большим в соответствии с ожиданиями, основанными на имеющихся видеороликах. Однако целевой объект может двигаться настолько быстро, что в следующем кадре окажется за пределами радиуса поиска. В этом случае трекер просто потеряет объект независимо от того, насколько точна его модель. Сбой из-за быстрого движения может быть особенно проблематичным для моделей, которые выполняют частые обновления, поскольку они будут использовать де-факто ошибочно классифицированные исправления для обновления модели трекера.

Еще одна проблема, связанная с быстрым движением цели, – размытие изображения при движении. Когда целевой объект движется очень быстро, матрица камеры получает изображения нескольких положений целевого объекта в течение одного кадра. Это приводит к эффекту усреднения, который размывает изображение и, что очень важно, сильно искажает внешний вид цели. В результате модель трекера, скорее всего, не захватит цель, потому что некоторые высокочастотные детали, необходимые для точного представления целевого объекта, внезапно исчезнут.

В последние годы разработчики моделей визуального отслеживания объектов пытались решить вышеупомянутые проблемы с помощью глубоких нейронных сетей и больших обучающих наборов. Далее мы рассмотрим основные подходы к глубокому обучению таких моделей.

10.2. КРАТКОСРОЧНОЕ ВИЗУАЛЬНОЕ ОТСЛЕЖИВАНИЕ ОБЪЕКТА

Прежде чем обсуждать появившиеся относительно недавно модели длительного отслеживания, мы сначала дадим краткое введение в краткосрочное отслеживание, которое долгое время было преобладающей парадигмой. Ме-

тоды краткосрочного отслеживания в первую очередь сосредоточены на визуальных экземплярах коротких эпизодов, обычно продолжительностью от 10 до 20 секунд. Основная задача краткосрочного отслеживания заключается в том, чтобы определять местонахождение цели с максимальной точностью как можно дольше, пока цель не будет потеряна. Как правило, краткосрочные трекеры не обладают механизмом восстановления, который может отследить цель после того, как она была потеряна, в основном из соображений вычислительной эффективности.

Краткосрочные трекеры оцениваются в основном по двум критериям: (1) точность определения координат цели и (2) быстродействие вывода. Общее мнение (Kristan et al., 2017) состоит в том, чтобы оценивать краткосрочные трекеры с упором либо исключительно на их точность, либо на точность с учетом быстродействия в реальном времени, и в этом случае скорость логического вывода должна соответствовать минимальному порогу. Такое ограничение зависит от конструкции оборудования и может варьироваться в зависимости от различных протоколов тестирования. Для случаев, когда отслеживание в реальном времени не является обязательным, существует несколько вариантов краткосрочного отслеживания, включая отслеживание только изображений RGB (Smeulders et al., 2014), использование изображений RGB-D (Zheng et al., 2017) или даже данные RGB, дополненные изображением с тепловизора (Li et al., 2019).

С методологической точки зрения существует несколько различных способов классификации методов краткосрочного отслеживания. Ввиду революции, вызванной глубоким обучением, и того факта, что большинство современных трекеров в настоящее время так или иначе полагаются на глубокую нейронную архитектуру, мы разделим трекеры на два семейства: неглубокие и глубокие. Категоризация неглубоких трекеров основана на основополагающей обзорной статье в (Smeulders et al., 2014).

10.2.1. Неглубокие трекеры

Неглубокие трекеры (shallow tracker) включают в себя большинство методов отслеживания, основанных на стандартных технологиях компьютерного зрения до появления глубокого обучения, которые перечислены далее.

Отслеживание путем поиска максимального подобия. Эта группа моделей определяет местоположение цели путем поиска максимального подобия¹ шаблона, построенного на основе предыдущих кадров, с различными областями-кандидатами искомого изображения. В эту категорию попадают несколько первых трекеров, использующих традиционные методы компьютерного зрения. К ним относятся методы, основанные на сопоставлении *нормализованной взаимной корреляции* (normalized cross-correlation, NCC) (Briechle, Hanebeck, 2001), *трекер Лукаса–Канаде* (KLT) (Baker, Matthews, 2004), *трекер Калмана* (Kalman appearance tracker, KAT) (Nguyen, Smeulders, 2004),

¹ Далее для краткости мы будем называть поиск максимального подобия (similarity matching) просто *сопоставлением*. – Прим. перев.

отслеживание среднего сдвига (mean-shift tracker, MST) (Comaniciu et al., 2000) и метод *локально неупорядоченного отслеживания* (locally orderless tracking, LOT) (Oron et al., 2015). Эти методы различаются в основном способами отбора областей-кандидатов и сопоставления с изображением шаблона. Например, NCC использует для сопоставления значения интенсивности в шаблоне и выполняет однородную выборку по искомому изображению.

Отслеживание путем сопоставления с расширенными моделями внешнего вида. Идея этого класса трекеров состоит в построении расширенной модели внешнего вида цели по предыдущим кадрам. Как правило, такие модели работают медленно, особенно потому, что в каждом кадре образцы-кандидаты сопоставляются с изображением шаблона, а также с кадрами, хранящимися в модели внешнего вида. Примером этого подхода является *инкрементное визуальное отслеживание* (incremental visual tracking, IVT) (Ross et al., 2008), где собственные изображения цели вычисляются с помощью инкрементного PCA по шаблону значения интенсивности цели. Они хранятся в «утекающей» памяти, где старые образы медленно забываются. Другими примерами этой группы являются отслеживание по аффинной группе (TAG) (Kwon et al., 2009) и отслеживание выборочных трекеров (TST) (Kwon and Lee, 2011).

Отслеживание путем сопоставления с ограничениями. Эта категория трекеров сокращает представление цели до разреженного представления. Для определения подходящей выборки-кандидата в кадрах поиска выполняется разреженная оптимизация. Такие методы в первую очередь ориентированы на сценарии, в которых внешний вид быстро меняется с течением времени и модель внешнего вида должна быстро адаптироваться. Для устранения дрейфа модели во время частых обновлений эти методы используют дополнительные ограничения, помимо традиционного отслеживания путем сопоставления.

Отслеживание с использованием дискриминативной классификации. Трекеры этой группы предлагают иной взгляд на проблему – они строят модель, основанную на различии объекта переднего плана по сравнению с фоном. Эти методы, также называемые отслеживанием путем обнаружения, создают классификатор, чтобы отличать целевые пиксели от пикселей фона, и обновляют классификатор на основе поступающих новых образцов. Старым, но популярным средством отслеживания из этой категории является средство отслеживания переднего плана и фона (Nguyen and Smeulders, 2006), в котором используются векторы признаков для дифференциации целевой области на локальном фоне. Другие аналогичные методы включают отслеживание по Хафу (Godec et al., 2013) и метод отслеживания, обучения и обнаружения (Kalal et al., 2010).

Отслеживание с использованием дискриминативной классификации с ограничениями. Для методов отслеживания, основанных на дискриминативной классификации, важно, чтобы выборки из цели и локального фона были правильно отобраны, иначе это может отрицательно сказаться на точности трекеров. Этот класс методов интегрирует процедуру маркировки в сам процесс обучения. Примером такой стратегии является *структурированное отслеживание вывода с ядрами* (structured output tracking with kernels, STR), когда новые обучающие данные выбираются из окружения положения цели

в предыдущем кадре. Во время обучения модель применяет ограничение, связанное с уверенностью в том, что выборка в исходном положении остается максимальной.

10.2.2. Глубокие трекеры

Трекеры на основе глубокого обучения, именуемые здесь *глубокими трекерами* (deep tracker), имеют много преимуществ по сравнению с их неглубокими, в основном созданными вручную аналогами. Это связано со способностью глубоких трекеров кодировать многоуровневую информацию и демонстрировать большую инвариантность и эквивариантность по отношению к изменениям внешнего вида цели. Существует несколько способов классификации глубоких трекеров. В некоторых публикациях их принято разделять на сиамские и дискриминативные. Далее мы воспользуемся обзорной статьей (Fiaz et al., 2019), чтобы сгруппировать различные трекеры в следующие две основные группы.

10.2.2.1. Отслеживание на основе корреляционного фильтра

Методы *отслеживания на основе корреляционных фильтров* (correlation filter-based tracking, CFT) для ограничения вычислительной стоимости выполняют вычисления, связанные с идентификацией цели, в частотной области. Они основаны на парадигме отслеживания путем обнаружения, а пример построения корреляционного фильтра с глубоким обучением показан на рис. 10.1. Целевой патч вырезается из изображения шаблона и используется для инициализации корреляционных фильтров в начале отслеживания. Для эффективного представления карты объектов строятся с использованием соответствующих методов извлечения. Первые методы вычисляли карту отклика с использованием поэлементного умножения между фильтром адаптивного обучения и извлеченными признаками, а также с использованием *дискретного преобразования Фурье* (ДПФ). Глубокое обучение позволяет кодировать эти признаки таким образом, что после взаимной корреляции достоверность достигается непосредственно в пространствен-

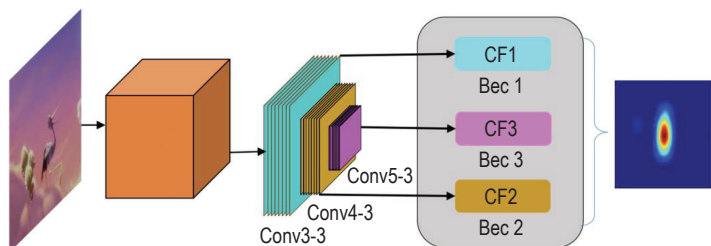


Рис. 10.1 ❖ Схематическое представление процесса обучения корреляционного фильтра с использованием функции кодирования CNN. Источник: Fiaz et al., 2019

ной области. Максимальный показатель достоверности указывает на новую позицию цели. Наконец, внешний вид цели в новом предсказанном местоположении обновляется путем извлечения признаков и обновления фильтров корреляции. В публикациях по отслеживанию существуют методы CFT, основанные на различных методах, и их можно описать следующим образом.

Простая корреляционная фильтрация (CFT). Основная особенность этого класса трекеров заключается в том, что они в той или иной форме используют ядерные корреляционные фильтры. В то время как ранние неглубокие трекеры для изучения этих ядер использовали обычные признаки, такие как HOG, названия цветов и т. д., глубокие трекеры используют рекуррентные или сверточные нейросети. В работе (Ma et al., 2015) предложили первый такой трекер, в котором использовались богатые возможности свертки с использованием сверточных фильтров, как показано на рис. 10.1. Он вычисляет независимые адаптивные корреляционные фильтры для каждого признака CNN и карты откликов. К улучшенным вариантам этого метода, среди прочих, относятся трекер на основе иерархических корреляционных признаков (Ma et al., 2015), глубокое отслеживание с хеджированием (Qi et al., 2016) и многозадачный корреляционный фильтр частиц (Zhang et al., 2017).

Регуляризованная корреляционная фильтрация (R-CFT). Трекеры на основе корреляционных фильтров сталкиваются с несколькими ограничениями, такими как необходимость в одинаковых размерах фильтра и патча, чувствительность обучения к негативным образцам и менее точные карты отклика при удалении от центра кадра. Трекеры R-CFT устраняют эту проблему за счет использования регуляризации процесса обучения фильтра. Например, метод *пространственно регуляризованного DCF* (spatially regularized DCF, SRDCF) (Danelljan et al., 2015) использует пространственную регуляризацию и во время отслеживания компонент регуляризации ослабляет фоновую информацию, тем самым делая трекер менее чувствительным к окружающему шуму. Другим примером является STRCF (Li et al., 2018), который использует дополнительную временную регуляризацию в SRDCF, чтобы избежать слишком резких скачков прогнозов, и допускает только плавные переходы на траектории отслеживания. Более свежим примером этого класса является ECO (Danelljan et al., 2017), который регуляризируется путем создания меньшего набора связей для эффективного захвата целевого представления с использованием матричной факторизации.

Корреляция на основе сиамской сети. Сиамская сеть объединяет два входа и дает один выход. Благодаря наличию двух подсетей она может изучать глубокие представления шаблона, а также искомого изображения. Этот класс трекеров сочетает в себе сиамские сети с CFT. Сиамские полностью сверточные сети (SiamFC) (Bertinetto et al., 2016) используют сверточное представление и слой корреляции для интеграции глубоких признаков, полученных из шаблона, а также из изображения-кандидата. Другими улучшенными вариантами таких методов являются трекеры SiamRPN (Li et al., 2018) и SiamRPN++ (Li et al., 2019), которые повышают точность отслеживания путем сопоставления при помощи сетей предсказания региона. Последней улучшенной версией является трекер Discriminative Model Prediction (DiMP) (Bhat et al.,

2019), который постоянно адаптирует представление шаблона для изучения улучшенного корреляционного фильтра из предыдущих кадров.

Корреляционная фильтрация на основе частей. В отличие от других CFT, эта категория трекеров изучает целевое представление по частям. Такие трекеры отслеживают по отдельности несколько частей изображения, и каждая часть может иметь свой корреляционный фильтр. Карты выходных откликов объединяются для создания окончательного отклика, по которому оценивается новое местоположение цели с использованием таких методов, как фильтрация частиц.

Корреляционная фильтрация на основе слияния. Точность трекеров может быть улучшена в определенных прикладных задачах за счет присоединения информации из дополнительных предметных областей (доменов). Это может быть слияние информации видимой и тепловой частей спектра и информации о глубине фокусировки из изображений или даже слияние низкоуровневых и высокоуровневых признаков глубокой сети. Например, метод *глубокого слияния признаков* (Wang et al., 2017) использует комбинацию *локальной обнаруживающей сети* (local detection network, LDN) и *глобальной обнаруживающей сети* (global detection network, GDN). LDN использует VGG-16 и объединяет информацию из разных частей сети для создания карты отклика. Если LDN не удастся обнаружить цель, в дело вступает GDN, параметры которой редко обновляются.

10.2.2.2. Отслеживание на основе некорреляционных фильтров

В методах *отслеживания на основе некорреляционных фильтров* (noncorrelation filter-based tracking, NCFT) не используются корреляционные фильтры, и они могут базироваться на концепциях изучения патчей, разреженности, суперпикселей, графов, сопоставления на основе частей или сиамского сопоставления.

Изучение патчей. Такие трекеры независимо извлекают информацию из разных частей изображения. Большинство подобных трекеров содержат комбинации общих слоев и слоев, специфичных для предметной области. В то время как общие слои используют обобщенное целевое представление всех последовательностей, уровень, специфичный для предметной области, отвечает за идентификацию цели с использованием бинарной классификации. Примерами глубоких трекеров из этой категории являются *структурозависимые сети* (structure aware network, SANet) (Fan, Ling, 2017) и *сверточные сети без обучения* (convolutional networks without training, CNT) (Zhang et al., 2016). SANet объединяет сверточные и рекуррентные признаки, используя стратегию конкатенации с пропуском для кодирования неразрезанной информации. CNT применяет иерархическую структуру с двумя слоями сверточной сети прямого распространения для точного создания представления цели. Нижний уровень извлекает локальные признаки, а глобальное представление цели формируется путем наложения простой ячеистой карты признаков, которая кодирует как локальную информацию, так и данные о геометрическом расположении.

На основе сиамских сетей. Сиамские сети также использовались для некорреляционной фильтрации. Их цель состоит в том, чтобы изучить представления, которые могут привести к сходству между заданными патчами. Как правило, шаблон и области поиска передаются в набор сверточных слоев, которые являются общими для двух подсетей сиамской модели, а глубокие признаки из этих двух подсетей затем объединяются в наборе последовательных полностью связанных слоев. Примерами являются SINT (Tao et al., 2016) и GOTURN (Held et al., 2016). Для более точной локализации цели выходные рамки SINT уточняются с использованием четырех регрессий ограничивающих рамок, обученных по ограничивающей рамке исходного кадра. Сиамские трекары, в том числе основанные на корреляции, будут подробно описаны позже, поскольку они особенно хорошо подходят для долгосрочного отслеживания.

На основе графов. Методы на основе обучения графа, как правило, используются для предсказания меток непомеченных вершин в графе. Некоторые современные трекары, такие как *древовидная CNN* (tree structure CNN, TCNN) (Nam et al., 2016), строят пространственные или временные графы для определения точных траекторий отслеживания. Существуют методы, которые используют для отслеживания как информацию, так и пространственно-временные графы. Для дальнейшего моделирования сложных отношений более высокого порядка структурозависимый трекар (Du et al., 2016) строит *гиперграфы* во временном измерении. Гиперграф строится с использованием частей-кандидатов в качестве узлов, а гиперребра обозначают отношения между частями.

10.3. ДОЛГОСРОЧНОЕ ВИЗУАЛЬНОЕ ОТСЛЕЖИВАНИЕ ОБЪЕКТА

Благодаря наличию стандартных наборов данных (Kristan et al., 2020; Wu et al., 2015; Smeulders et al., 2014) за последние несколько лет технология визуального отслеживания значительно продвинулась вперед. Эти наборы данных в основном были разработаны для решения проблем, возникающих при краткосрочном отслеживании. Например, средняя продолжительность видео в ALOV (Smeulders et al., 2014) и OTB (Wu et al., 2015) составляет всего около 10 и 20 секунд соответственно. Назначение этих наборов данных заключалось в том, чтобы представить трудные моменты, такие как изменения условий освещения, резкое движение, беспорядок, большие деформации, внезапные окклюзии и др.

Однако при работе с более длинными видео возникают дополнительные проблемы и требования, помимо тривиального условия, что модель должна выполнять прогнозы для более длительного времени. Ранее мы рассмотрели различные проблемы с отслеживанием и сгруппировали их в задачи по критериям внешнего вида, обучения и технических проблем. Все проблемы, которые существуют для краткосрочных трекаров, справедливы и для

долгосрочных. Среди всех проблем есть две, которые занимают особое место в долгосрочном отслеживании, – это устаревание модели и исчезновение и повторное появление цели. Хотя эти две проблемы не являются уникальными для длительного отслеживания, их последствия гораздо более сильно и нетривиально выражены в трекерах, которые работают с более длинными последовательностями.

10.3.1. Устаревание модели при длительном отслеживании

Хорошие результаты отслеживания на тестовых наборах данных часто интерпретируют как решение всех основных проблем отслеживания. На практике, однако, возникают специфические проблемы, когда продолжительность отслеживания превышает, допустим, полчаса. При практическом применении отслеживания длинные видео встречаются гораздо чаще, чем короткие, например в различных взаимодействиях между людьми, спортивных репортажах, документальной съемке и телешоу. Однако слишком продолжительное обновление может в конечном итоге разрушить встроенную модель трекера и привести к потере цели. В то время как устаревание модели может быть незаметным в краткосрочных видео, накопленный эффект очень заметен при длительном отслеживании.

Мы продемонстрируем это на примере синтетического эксперимента, изображенного на рис. 10.2. Чтобы показать серьезное влияние устаревания модели при обработке длинного видеоряда, мы случайным образом выбираем видеоролик из набора данных OTB50 (Wu et al., 2015) и искусственно расширяем его в соответствии со следующим периодическим законом:

$$x' = [x_1, \dots, x_{T-1}, x_T, x_T, x_{T-1}, \dots, x_2, x_1, x_1, x_2, \dots, x_T, \dots]. \quad (10.4)$$

Расширение видео таким способом гарантирует, что любые различия в точности отслеживания, наблюдаемые в более поздних частях длинной последовательности, будут вызваны исключительно увеличением длины видео, а не дополнительными визуальными затруднениями. На рис. 10.2 мы видим прогнозы ECO (Danelljan et al., 2017) – одного из самых успешных несиамских трекеров, который опирается на частые обновления, – для трех разных кадров, наблюдаемых в трех разных повторениях исходного видео. Мы видим, что при увеличении числа повторений фрагментов предсказания трекера со временем становятся все менее точными, хотя кадры абсолютно идентичны. Причина в устаревании модели, вызванном постепенным, но ошибочным и интенсивным обновлением. Дрейф трекера, вызванный устареванием модели, давно известен (Smeulders et al., 2014), но в контексте краткосрочного отслеживания этот вопрос не очень актуален. При долгосрочном отслеживании устаревание модели чаще всего приводит к катастрофическим последствиям, даже если при каждом обновлении модели допускаются всего лишь небольшие ошибки, как показано в исследовании (Gavves et al., 2020).

Чтобы разработать меры борьбы с этим негативным эффектом длительного отслеживания, необходимо теоретическое обоснование, способное дать математическое определение лежащему в основе данного явления механизму.

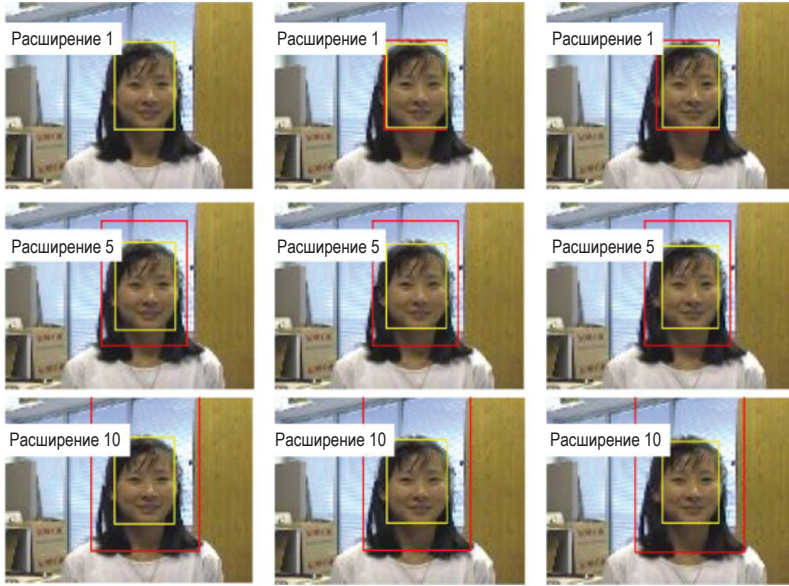


Рис. 10.2 ❖ Прогнозы трекера ECO (Danelljan et al., 2017) для искусственно расширенного видео, созданного на основе данных OTB50 (красный прямоугольник: прогноз трекера, желтый прямоугольник: эталонный прогноз). Здесь очевидно преобладает устаревание модели, хотя вариация внешнего вида остается неизменной. Из-за большого количества обновлений устаревание модели заметно с самых ранних стадий, даже для четко видимых целевых объектов, движущихся медленно. Источник: Gavves et al., 2020

Расширяя математическое определение трекеров и предполагая, что мы обновляем параметры модели в каждом кадре, мы получаем следующие уравнения:

$$\phi_{t+1} = \underset{\phi}{\operatorname{argmin}} \mathcal{L}(x_{1:t}, y_{1:t}), \quad (10.5)$$

$$y_{t+1} = f(x_{t+1}, \phi_{t+1}), \quad (10.6)$$

где f – модель трекера с параметрами ϕ , которая минимизирует потери \mathcal{L} трекера по набору данных $D = [x_{1:t}, y_{1:t}]$ на такте $t + 1$. Набор данных состоит из кадров $x_{1:t} = [x_1, \dots, x_t]$, а модель трекера f возвращает в качестве выходных данных предсказания ограничивающей рамки $y_{1:t} = [y_1, \dots, y_t]$. Чтобы упростить и упорядочить обозначения, мы используем $f_{i,t}$ для обозначения выходных данных модели трекера с параметрами ϕ_i , примененной к кадру x_i . В простейшем случае параметры модели обновляются путем небольших

шагов в направлении градиента поверхности потерь, а именно с использованием подхода градиентного спуска (или его вариантов):

$$\frac{\partial \phi}{\partial t} = -\eta \nabla_{\phi} \mathcal{L}_t; \quad (10.7)$$

$$\phi_{t+1} = \phi_t - \eta \nabla_{\phi} \mathcal{L}_t. \quad (10.8)$$

Таким образом, центральное место в задаче обучения отслеживающей модели занимает градиент потерь при отслеживании, обусловленный параметрами модели. Продолжая на $\nabla_{\phi} \mathcal{L}$ и используя математическое ожидание по t временным шагам $\mathbb{E}[\cdot] = \frac{1}{t} \sum_{i=1}^t [\cdot]$, имеем:

$$\nabla_{\phi} \mathcal{L}_t = \nabla_{\phi} \mathbb{E}[(y_i - f_{i,t})^2] \quad (10.9)$$

$$= 2\mathbb{E}[f_t \nabla_{\phi} f_t] - 2\mathbb{E}[y_t \nabla_{\phi} f_t]. \quad (10.10)$$

Чтобы перейти от уравнения (10.9)–(10.10), мы исходим из того, что координаты ограничивающей рамки y_i , предсказанные в предыдущих кадрах, становятся входными переменными с постоянными значениями. Таким образом, они не зависят от ϕ , а $\mathbb{E}[\nabla_{\phi} y_i^2] = 0$. Это сильное предположение, учитывая, что на практике y_i определяются моделью с параметрами ϕ .

Подставляя уравнение (10.10) в (10.8), обновление параметров модели можно описать как

$$\phi_{t+1} - \phi_t = -2\eta [\mathbb{E}[f_{i,t} \nabla_{\phi} f_{i,t}] - \mathbb{E}[y_i \nabla_{\phi} f_{i,t}]]. \quad (10.11)$$

Интересная, но часто упускаемая из виду реальность заключается в том, что хотя отслеживание рассматривается как задача обучения с учителем, в наборе обучающих данных есть только одна выборка данных, которая определенно верна. Эта единственная эталонная выборка – пара (x_1, y_1^*) , определенная пользователем в первом кадре, где y_1^* представляет собой координаты заданной пользователем ограничивающей рамки, описывающей объект. Несмотря на то что все остальные ограничивающие рамки $y_i > 1$ используются для переобучения и тонкой настройки трекера, нет никакой гарантии, что эти рамки действительно верны или хотя бы достаточно хороши для обучения. На самом деле если бы предсказания y_i были гарантированно хороши для переобучения трекера, то переобучение не потребовалось бы.

На основании вышеупомянутого аргумента разумно ожидать, что прогнозы, которые также служат будущими обучающими выборками для переобучения трекера, являются зашумленными измерениями истинных координат ограничивающей рамки y_i^* . Предполагая, что имеем дело с гауссовым шумом с дисперсией σ_i^2 , мы можем записать следующее уравнение:

$$y_i = y_i^* + \delta_i \text{ и } \delta_i \sim N(0, \sigma_i^2). \quad (10.12)$$

Подставив уравнение (10.12) в (10.11), после перестановки членов имеем:

$$\phi_{t+1} - \phi_t = -2\eta[\mathbb{E}[f_{i,t} \nabla_{\phi} f_{i,t}] - \mathbb{E}[y_i^* + \delta_i] \nabla_{\phi} f_{i,t}] \quad (10.13)$$

$$= \underbrace{-2\eta\mathbb{E}[(f_{i,t} - y_i^*) \cdot \nabla_{\phi} f_{i,t}]}_{\text{Идеальное обновление параметра}} + \underbrace{2\eta\mathbb{E}[\delta_i \cdot \nabla_{\phi} f_{i,t}]}_{\text{Смещение параметра}}. \quad (10.14)$$

Обновление параметров трекера состоит из двух компонентов. Первый член в уравнении (10.14) соответствует компоненту обновления идеальной модели, поскольку он исправляет ошибку, сделанную предсказанием модели $f_{i,t}$ по сравнению с идеальной эталонной рамкой y_i^* . Второй член представляет собой смещение параметра, так как он напрямую зависит от ошибки, сделанной прошлыми предсказаниями δ_i . При $\delta_i = 0$ ошибки бы не было и обновления параметров тоже были бы идеальными.

Динамика модели. Вычислив влияние прошлых ошибок на обновления параметров трекера, мы можем затем исследовать влияние на динамику модели $\frac{\partial f}{\partial t}$ с течением времени. В частности, после обновления параметров связь между прошлой моделью $f_{i,t}$ и следующей $f_{i,t+1}$ имеет вид:

$$\frac{\partial f}{\partial t} \propto f_{i,t+1} - f_{i,t} = \frac{\partial f}{\partial \phi} \frac{\partial \phi}{\partial t} \Rightarrow \quad (10.15)$$

$$f_{i,t+1} = f_{i,t} + \frac{\partial f}{\partial \phi} \frac{\partial \phi}{\partial t}. \quad (10.16)$$

Поскольку $\frac{\partial \phi}{\partial t} \propto \phi_{t+1} - \phi_t$, комбинируя уравнения (10.16) и (10.14), получаем:

$$f_{i,t+1} = f_{i,t} + \underbrace{-2\eta\mathbb{E}[(f_{i,t} - y_i^*) \cdot \|\nabla_{\phi} f_{i,t}\|^2]}_{\text{Идеальное обновление модели}} + \underbrace{2\eta\mathbb{E}[\delta_i \cdot \|\nabla_{\phi} f_{i,t}\|^2]}_{\text{Устаревание модели}}. \quad (10.17)$$

Из уравнения (10.17) можно сделать следующий вывод. Из-за непрерывных обновлений модель трекера корректирует свои прогнозы на величину, которая линейно пропорциональна прошлым ошибкам. Мы называем эту величину распадом модели.

Длительное отслеживание и устаревание модели. Поскольку динамика модели в уравнении (10.17) является рекурсивной, подразумевается, что член смещения накапливается и фактически ухудшается со временем. Так как кумулятивное устаревание модели достаточно мало, обычно на ранних итерациях динамика модели является достаточно точной. Ранние ошибки δ_i малы не только потому, что трекер все еще точен, но и потому, что количество слагаемых t мало. По этой причине устаревание модели не является проблемой и часто остается незамеченным в коротких видеороликах. Однако в более длинных видео, где t и количество слагаемых растут, ошибки $\delta_{i,t}$ также растут, и кумулятивное устаревание модели становится заметным.

10.3.2. Исчезновение и повторное появление цели

Предполагая, что цель никогда не исчезает, краткосрочные трекары могут всегда возвращать свой наиболее вероятный прогноз, независимо от того, является ли вероятность прогноза высокой или низкой. Поэтому обученный классификатор подобного краткосрочного трекера не требует минимальной оценки правдоподобия, прежде чем объявить, что определенное место содержит целевой объект. В свою очередь, это означает, что обученному классификатору не нужно калибровать свою достоверность.

Однако в более длинных видеопоследовательностях цель, скорее всего, исчезнет из кадра и снова появится. Часто это случается несколько раз. Значит, любой долгосрочный трекер должен уметь моделировать отсутствие цели в кадре. Хотя это кажется небольшим отличием от краткосрочного трекера, на самом деле оно может иметь важные теоретические и практические последствия для моделирования. Прежде всего модель трекера должна уметь различать, почему объект стал невидимым для нее – из-за исчезновения или из-за серьезных изменений внешнего вида. Более того, если модель использует обновления для учета изменений внешнего вида объекта, функция подобия модели будет динамически изменяться со временем. Это означает, что нельзя зафиксировать заданный минимальный порог обнаружения. Вместо этого порог должен быть либо определен динамически, либо модель должна откалибровать свои прогнозы таким образом, чтобы минимальный порог обнаружения оставался действительным. Наконец, при наличии долгосрочного устаревания модели динамическая адаптация порога может быть затруднена без прямого влияния на точность трекера и дополнительного смещения модели, особенно в случае последовательных обновлений.

10.3.3. Долгосрочные трекары

Долгосрочные трекары должны не только решать общие проблемы слежения, с которыми сталкиваются краткосрочные трекары, но и дополнительно обрабатывать долгосрочное затухание, а также исчезновение и повторное появление цели. Теоретически устаревание модели неизбежно, когда к краткосрочным трекерам добавляется последовательное обучение. Решение проблемы долгосрочного устаревания привлекает наибольшее внимание из-за ее фундаментального и катастрофического характера. Мы выделяем три семейства подходов долгосрочного отслеживания, сгруппированных по типам обучения и обновления модели трекера с учетом долгосрочного устаревания: предварительное обучение, последовательное обучение и гибридное обучение.

10.3.3.1. Предварительное обучение и сиамские трекары

В основе всех алгоритмов отслеживания лежит модельная функция $f_\phi: \mathcal{Y}_1 \times \mathcal{X}_1 \times \mathcal{X}_t \rightarrow \mathcal{Y}_t$ из уравнения (10.1), согласно которой изображение цели сопоставляется с поступающими кадрами. Модель трекера возвращает прогнозы $y_{1:t}$, которые, как мы надеемся, достаточно близки к истинным местоположени-

ям $y_{1:t}^*$. Идеальная функция сопоставления для отслеживания обеспечивает правильное сопоставление, даже если цель на видео частично перекрыта другими предметами, меняет свой масштаб, вращается в плоскости и вне ее или подвергается неравномерному освещению, движению камеры и другим мешающим факторам (Smeulders et al., 2014; Wu et al., 2015).

Как мы упоминали ранее, для решения этих проблем с отслеживанием модели часто полагаются на последовательные обновления, используя прогнозы модели в качестве псевдоэталона. Это, в свою очередь, приводит к ухудшению модели, которое особенно сильно выражено в случае длинных и сложных видео, поскольку мы предполагаем, что прогнозы модели эквивалентны эталонным прогнозам во время обновлений: $y_{1:(t-1)} \equiv y_{1:(t-1)}^*$. Этим предположением мы признаем следующий специфический парадокс, который называем *парадоксом отслеживания* (tracking paradox). С одной стороны, если модель настолько точна, что можно предположить $\hat{y}_{1:(t-1)} \equiv y_{1:(t-1)}$, то модель не требует дальнейшего переобучения. Другими словами, переобучение не дает дополнительных преимуществ, поскольку предсказания модели уже совершенны. С другой стороны, если модель не так точна, как предполагалось, принимая $\hat{y}_{1:(t-1)} \equiv y_{1:(t-1)}$, мы получаем только временное улучшение точности, но постепенно делаем модель все хуже и хуже и в конце концов доводим ее до полного разрушения.

Чтобы разрешить парадокс, мы могли бы взять модель трекера, которая *не требует обновления*. В этом случае модель должна иметь возможность определять местонахождение цели просто по информации, которая была доступна до отслеживания, то есть только по ограничивающей рамке в первом кадре. Это явно затруднительно, так как в сложных видео существуют серьезные проблемы, к которым модель трекера должна быть невосприимчивой, как было сказано в разделе 10.1.1. Кроме непрерывного обновления, традиционный способ решения этих проблем заключается в явном моделировании каждого из вышеупомянутых искажений путем введения аффинных преобразований (Lucas, Kanade, 1981), вероятностного сопоставления (Comaniciu et al., 2000), собственных изображений (Ross et al. al., 2008), инвариантов освещения (Nguyen, Smeulders, 2006), обнаружения окклюзии (Pan and Hu, 2007) и т. д. К сожалению, явное моделирование отдельных проблем – при игнорировании всех остальных – дает трекары, которые могут быть оптимальными для одного типа искажений, но неоптимальными для многих других, что приводит к ухудшению точности.

Вместо того чтобы явно моделировать все возможные искажения с помощью модели трекера f , можно представить отслеживание как проблему сопоставления изображений, где искомое изображение всегда является целью в первом кадре. Поскольку запрос остается фиксированным и не меняется из-за последовательных обновлений с использованием несовершенных прогнозов отслеживания, модель гарантированно будет работать стабильно и надежно в течение любого периода времени. Определение модели трекера приобретает следующий вид:

$$y_{t+1} = \underset{y}{\operatorname{argmin}} f(h_\phi(\mathbf{z}), h_\phi(\mathbf{x}_t[y])). \quad (10.18)$$

В этом определении \mathbf{z} – участок целевого объекта в первом кадре $\mathbf{x}_t[y]$ соответствует патчу в t -м кадре, который, в свою очередь, соответствует координатам ограничивающей рамки y , h_ϕ – сверточные нейронные сети, вычисляющие представления из \mathbf{z} и $\mathbf{x}_t[y]$, а f – функция поиска максимального подобия трекера. Каких-либо ограничений по типу сетей h_ϕ нет, хотя на практике одни сети работают лучше других (Tao et al., 2016; Li et al., 2019). Хотя для простоты мы используем одни и те же параметры ϕ для обеих нейронных сетей, параметры не обязательно должны быть общими (Li et al., 2019).

В случае отслеживания путем поиска максимального подобия (сопоставления) целей критическим компонентом модели является функция сопоставления, которая должна быть устойчива ко всем нежелательным проблемам отслеживания. Традиционно эта функция сопоставления обучается в последовательном режиме применительно к конкретной цели \mathbf{z} . Альтернативой является внешнее обучение функции сопоставления в автономном режиме, в частности путем сравнения появлений объектов, записанных в разные моменты времени, то есть (x_i, x_j, y_{ij}^*) , где $y_{ij}^* = 1$, если x_i и x_j представляют тот же объект, и -1 в противном случае. Функция сопоставления f может быть реализована с помощью многослойного персептрона (полностью связанный слой) и обучена для минимизации контрастных потерь (Tao et al., 2016):

$$\operatorname{argmin}_{\phi} \frac{1}{2} y_{ij}^* d_{ij}^2 + \frac{1}{2} (1 - y_{ij}^*) \max(0, \epsilon - d_{ij}^2), \quad (10.19)$$

$$d_{ij} = \|h_\phi(x_i) - h_\phi(x_j)\|^2, \quad (10.20)$$

или с использованием косинусного подобия для минимизации логистических потерь (Bertinetto et al., 2016):

$$\operatorname{argmin}_{\phi} \log \left(1 + \exp \left(-y_{ij}^* h_\phi(x_i)^T h_\phi(x_j) \right) \right). \quad (10.21)$$

По сути, для обеих потерь расстояния, будь то евклидова норма или косинусное подобие, являются внутренними произведениями. Учитывая, что при минимизации поиска в уравнении (10.18) мы перебираем местоположения ограничивающей рамки, которые плотно упакованы поверх карт признаков, итерационные внутренние произведения могут быть более компактно записаны как свертки, то есть

$$y_{t+1} = \operatorname{argmin}_y h_\phi(\mathbf{z}) * h_\phi(\mathbf{x}_t[y]). \quad (10.22)$$

Уравнение (10.22) позволяет очень эффективно использовать полностью сверточные сиамские сети (Bertinetto et al., 2016).

Визуальные трекары, которые отслеживают объекты с использованием сиамских глубоких нейронных сетей, называются *сиамскими трекарами* (рис. 10.3). Сиамские трекары имеют две ветви сверточных нейронных сетей. Первая из них – это ветвь шаблона, которая сворачивает целевой патч так, как это определено в первом кадре. Результатом является шаблон, с помощью которого каждый будущий кадр может быть дополнительно свернут

для определения положения цели. Вторая – это ветвь кандидатов, обрабатывающая новые кадры в видеопоследовательностях и пытающаяся найти объект, максимально подобный цели.

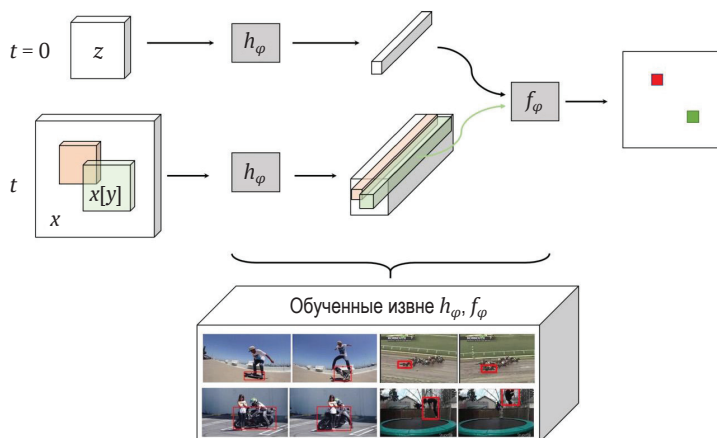


Рис. 10.3 ❖ Сиамский трекер состоит из двух ветвей, каждая из которых смоделирована сверточной нейронной сетью. Первая ветвь всегда содержит патч целевого объекта в момент $t = 0$. Вторая ветвь анализирует любой другой кадр в видео. Затем представления двух ветвей сравниваются при помощи функции подобия. Функция подобия обучается автономно с использованием отслеживаемых объектов из разных наборов данных. Хотя эта функция не видела будущие цели, она все же может точно оценить подобие их внешнего вида для выполнения отслеживания путем поиска максимального подобия. Схема основана на работах Bertinetto et al. (2016); Tao et al. (2016)

Область поиска. Как и в случае краткосрочных трекеров, вместо поиска цели по всему кадру можно ограничить модель поиском следующих местоположений цели в пределах заданного радиуса ρ . Ограничение области поиска помогает повысить вычислительную эффективность алгоритма трекера, поскольку он должен анализировать гораздо меньшую площадь кадра. Ограничение поиска определенной областью также может снизить вероятность ложных срабатываний, которые могут случайно возникнуть при просмотре всего изображения.

Радиус r_0 – гиперпараметр, зависящий от максимальной скорости цели. Как и в большинстве случаев, мы не можем знать максимальную скорость цели. Поэтому если мы решили задать область поиска, то должны делать это осторожно. Если скорость перемещения цели превышает $r_0/\Delta t$, где Δt – интервал между любыми двумя кадрами (величина, обратная скорости записи в кадрах в секунду), то цель окажется вне области поиска и трекер ее потеряет. Если радиус r_0 задать таким, что $r_0 = \max(H, W)$, где H и W – высота и ширина кадра, то трекер будет просматривать весь кадр.

При определении размера области поиска необходимо балансировать между повышением точности и ухудшением полноты отклика. Задав небольшую область поиска, можно повысить точность, избегая избыточных

ложных срабатываний. Однако небольшая область поиска также может ухудшить полноту отклика, поскольку все местоположения за пределами области поиска автоматически помечаются как отрицательные; в случаях, когда цель оказалась за пределами области поиска, это будут ложноотрицательные результаты. Аналогичным образом можно привести обратные аргументы для больших областей поиска¹.

Преимущества сиамских трекеров. Сиамские сети имеют определенные преимущества. По большому счету, самым важным преимуществом является возможность избежать устаревания модели из-за отсутствия обновлений модели на протяжении всего видео. Причина в том, что, несмотря на ошибки прогнозирования δ_i , модель никогда не обновляется, поэтому $\nabla_{\phi} f_{i,t} = 0$ (раздел 10.3.1). Разработчику модели не нужно беспокоиться о том, что модель рухнет и больше не восстановится. Это делает сиамские трекеры идеальным инструментом для долгосрочного отслеживания.

Если сравнивать сиамские трекеры на сверточных нейронных сетях (Tao et al., 2016) с полностью сверточными сиамскими трекерами (Bertinetto et al., 2016b), то первые, как правило, имеют значительно более высокую точность и лучшую надежность за счет более высоких вычислительных затрат. Причина в том, что сверточные нейронные сети с полностью связанными слоями могут быть предварительно интенсивно обучены на базах данных изображений, таких как ImageNet (Russakovsky et al., 2015), и, таким образом, дополняют сиамскую модель сильными априорными знаниями. Напротив, из-за отсутствия полносвязных слоев полностью сверточные нейронные сети обладают значительно более высокой эффективностью и производительностью в реальном времени, но меньшей дискриминативной способностью (Valmadre et al., 2018).

Полагаясь исключительно на предварительно обученную сквозным методом функцию подобию, сиамские трекеры требуют только начального обучения и не нуждаются в последовательном обучении во время работы. Это означает, что мы можем заранее выполнить автономное обучение, чтобы изучить любой тип инвариантов, которые понадобятся трекеру. Мы можем сделать это во время обучения, просто предоставив функции подобию примеры вариантов, которые хотим сопоставить друг с другом. Более того, поскольку обучение проводится в автономном режиме, мы можем использовать очень большие наборы данных и повышать точность без увеличения вычислительных затрат или затрат на память.

Еще одним преимуществом сиамских трекеров является то, что функцию подобию можно обучить для оптимального объединения карт признаков из разных слоев. Карты признаков из разных слоев фиксируют различные типы особенностей объекта: более ранние слои фиксируют низкоуровневые геометрические паттерны, такие как края или углы, а более поздние фиксируют семантические паттерны высокого уровня, которые идентифицируют лица

¹ Точность (precision) вычисляется как отношение числа истинных результатов к сумме истинных и ложных результатов. Полнота отклика (recall) вычисляется как отношение числа истинно положительных результатов к сумме истинно положительных и ложноотрицательных результатов.

или типы объектов. Следовательно, при поиске совпадения сиамские трееры по своей сути используют как низкоуровневую геометрическую, так и высокоуровневую семантическую информацию, что позволяет выполнять детальный поиск по новым кадрам. Это свойство особенно важно в случае вводящего в заблуждение фона, когда объекты могут выглядеть одинаково, но иметь различную семантику – такая ситуация запутывает модели, которые могут использовать для устранения неоднозначности объектов данные только из одной и той же последовательности.

Особенностью сиамских трееров является то, что при отслеживании они игнорируют время, если не определена область поиска и модель ищет объект по всему кадру. Причина в том, что в алгоритме нет переменной времени: внешний вид целевого объекта задается в первом кадре, модель поиска максимального подобия обучается в автономном режиме и используется на протяжении всего процесса отслеживания без изменений на промежуточных кадрах и прогнозах, и область поиска по каждому новому кадру также не зависит от предыдущего местоположения цели. Хотя это свойство может показаться нелогичным, оно не мешает отслеживанию, если функция подобия может почти идеально сопоставлять внешний вид разных представлений одного и того же объекта. Более того, если задать в качестве области поиска весь кадр, то даже если цель упущена в одном кадре, у модели остается возможность повторно найти местоположение цели в последующих кадрах.

Ситуации, ведущие к сбою отслеживания. Популярность сиамских трееров постоянно растет благодаря хорошей точности и замечательной устойчивости к дрейфу модели. Однако в их архитектуре кроются некоторые недостатки, которые дают о себе знать в определенных сценариях.

Самый очевидный недостаток базовой архитектуры сиамских трееров – путаница при наличии множества однотипных объектов. Эта путаница особенно заметна, когда объекты не просто похожи, но идентичны, например при попытке отследить одного человека в марширующем оркестре. В то время как отследить одного конкретного человека в марширующем оркестре, пожалуй, сложно даже для человека, сиамский треер по своей природе не может различать идентичные или почти идентичные объекты. Причина в том, что в простых сиамских треерах отсутствует компонент моделирования движения.

Еще одной проблемой для сиамских трееров является полное отсутствие обновлений моделей. Даже если мы предположим, что функция подобия идеальна, бывают случаи, когда целевой объект меняет внешний вид либо сам по себе (например, пешеход меняет одежду), либо под влиянием окружающей среды (например, значительные изменения освещения, которые могут изменить внешний вид объекта). В этом случае функция подобия делает больший упор на форму объекта-кандидата, который должен быть в основном похож на форму исходного объекта, а также на семантику объекта, которая не меняется. Это неплохое свойство, учитывая, что сверточные нейронные сети продемонстрировали большую способность к обобщению при распознавании объектов. Действительно, благодаря этому обобщению сиамские трееры, такие как (Taο et al., 2016), продемонстрировали устойчивость к обычным проблемам

отслеживания (Valmadre et al., 2018). Тем не менее это обобщение увеличивает вероятность путаницы при наличии нескольких похожих объектов.

Наконец, поскольку сиамские трекары основаны на сверточных нейронных сетях, они сталкиваются с теми же проблемами, что и базовые сети. В частности, стандартные сверточные нейронные сети не являются строго инвариантными или эквивариантными по отношению к изменениям масштаба и поворотам, будь то повороты в плоскости или вне плоскости. Для устранения зависимости от вариаций масштаба и поворота применяют тщательное предварительное обучение сиамских сетей, увеличение объема данных или специальные стратегии постобработки, такие как повторение поиска в нескольких масштабах или поворотах. Эти подходы являются дорогостоящими и работают только до тех пор, пока вариации в достаточной степени представлены в данных и их дополнениях.

10.3.4. Инвариантность и эквивариантность представления

На входные данные всегда влияет шум, и, следовательно, нет двух абсолютно одинаковых точек данных. Например, никогда не бывает двух абсолютно идентичных изображений, поскольку объект мог изменить свое местоположение, свой внешний вид или ракурс, освещение в сцене могло стать другим и т. д. Некоторые из этих вариаций не имеют отношения к поставленной задаче прогнозирования, то есть идентификации и локализации объекта в случае визуального отслеживания. Например, не имеет значения, как объект освещен и падает ли на него тень; трекарь должен иметь возможность локализовать целевой объект в любом случае. Однако некоторые из этих изменений могут иметь значение. Например, изменение масштаба объекта в кадре не имеет отношения к идентификации, но может иметь значение для вывода о том, что объект движется к камере или удаляется от нее. И конечно же, понимание различных вариаций одного и того же объекта на самом деле имеет решающее значение для обобщения.

Представления *инвариантны* к определенным изменениям, если на них не влияет наличие этих изменений. Инвариантность к освещению, например, означает, что представление будет в значительной степени одним и тем же при изменении освещения. Представления *эквивариантны* к определенным изменениям, когда представления меняются пропорционально изменению объекта или его среды. Говоря упрощенно, эквивариантность к вращению означает, что поворот объекта на $\pi/4$ оказывает в два раза большее влияние на его представление, чем поворот на $\pi/8$. Далее мы опишем, как современные средства визуального отслеживания объектов включают инвариантность и эквивариантность в свои представления.

10.3.4.1. Инвариантность при отслеживании

В сиамских трекарах функция подобия имеет первостепенное значение. Причина в том, что трекарь полагается исключительно на эту функцию, чтобы

сравнить два произвольных патча и заключить, изображают ли они один и тот же целевой объект или нет, независимо от любых возможных изменений во внешнем виде. Однако ожидается, что во время просмотра видео объект может испытывать серьезные искажения и изменения внешнего вида. Эти помехи могут быть вызваны либо самим объектом (человек снял куртку или автомобиль встал под углом, отличным от того, который наблюдался на первом кадре), либо окружающей средой (например, перекрытие другим объектом, изменение освещения или тень от близлежащего объекта). Функция подобия должна безошибочно определять, что объект остается тем же самым, несмотря на все подобные помехи. То есть функция подобия должна научиться быть инвариантной ко всем возмущениям, обычно встречающимся при отслеживании, как мы описали их в разделе 10.1.1.

Инвариантность может быть достигнута либо путем жесткого программирования в коде алгоритма, либо путем изучения возможных возмущений на основе данных. В первом случае метод жестко настроен на игнорирование определенных изменений визуальных входных данных, например поворота (Gupta et al., 2021) или масштаба объекта (Sosnovik et al., 2021). Во втором случае метод наблюдает за различными вариациями больших наборов данных и учится игнорировать те вариации, которые не имеют отношения к сопоставлению. Преимущество инвариантности с жестким кодированием заключается в том, что можно достичь желаемого с гораздо меньшими наборами данных. Напротив, инвариантность на основе обучения требует гораздо больших наборов данных. Причина в том, что модель должна улавливать все возможные вариации внешнего вида, относящиеся к этим инвариантностям, и затем обобщать их.

Жесткое программирование позволяет уменьшить размер модели, но в общем случае результат зависит от математической модели инвариантности. Более того, если математическая модель инвариантности не делает никаких предположений или делает лишь несколько предположений, инвариантности распознаются в данных более точно. С другой стороны, инвариантности на основе обучения не требуют явного математического моделирования указанных инвариантностей. Это намного удобнее, когда смешивается несколько разных изменений, что часто случается при отслеживании данных. Кроме того, поскольку не требуется явная математическая модель, инвариантные модели на основе обучения обычно проще, и нужно сосредоточиться только на сборе достаточно больших и разнообразных данных.

Базовые сиамские трекеры (Tao et al., 2016; Bertinetto et al., 2016) используют последний вариант и эффективно учатся на данных игнорировать инвариантности, которые не имеют отношения к идентификации целевого объекта. Например, после обучения на многочисленных образцах объектов, которые претерпели деформацию или изменения освещенности, сиамские трекеры предсказывают, изображают ли какие-либо два патча один и тот же объект или нет, если видеоряд содержит те или иные деформации либо изменения освещенности. Это означает, что для обучения инвариантному сопоставлению сиамским трекерам нужны масштабные и разнообразные наборы обучающих данных.

Использование больших наборов данных непосредственно во время отслеживания невозможно. Следовательно, функция сходства сиамского трекера

должна быть обучена в автономном режиме на внешних данных. Автономный режим означает, что обучение не происходит одновременно с отслеживанием. Внешние данные означают, что обучение не использует целевые данные, которые пользователь определил бы в первом кадре видео, так как алгоритм трекера не может иметь доступа к таким данным. Поэтому функция подобию должна быть обучена на внешних данных и до развертывания трекера.

Автономное обучение на внешних данных резко отличается от подхода, применяемого в большинстве традиционных трекеров, чья функция подобию обычно изучается последовательно в ходе отслеживания. Последовательное обучение трекера в рабочем режиме означает, что он использует только внешний вид цели на первом кадре (единичный положительный образец), прогнозы трекера по всему видео (псевдоположительные образцы), а также внешний вид фона (отрицательные образцы) и, возможно, другие появления объектов из разных видео (отрицательные образцы).

Чтобы изучить инвариантность на основе данных, требуется большой набор видеороликов движущихся объектов из разных категорий. Кроме того, эти отслеживаемые объекты должны быть снабжены аннотациями с указанием их местоположения на протяжении всего видео. Чтобы создать набор данных, после этого нужно просто собрать все возможные комбинации патчей (x_i, x_j) в моменты времени i и j . Пары (x_i, x_j) должны подвергаться изменениям, которые нужно смоделировать, чтобы научиться распознавать их или игнорировать. Другими словами, ролики должны быть разноплановыми и не содержать похожих объектов и сценариев, иначе модель не сможет обобщать.

Важным моментом является то, что во время обучения не должно быть абсолютно никаких совпадений между видео, используемыми для внешнего обучения, и любым из видео для рабочего отслеживания. А именно нельзя допускать попадания в автономные обучающие данные ни одной из реальных целей отслеживания, так как в этом случае модель по существу превратится в детектор объекта. Вместо этого при автономном обучении модель должна сосредоточиться на изучении общего набора помех и изменений, а не на характерных для объекта шаблонах. Как только формирование функции сопоставления с помощью внешних данных завершено, она больше не адаптируется и применяется к ранее не встречавшимся объектам и видео как есть.

Сиамское отслеживание напоминает парадигму поиска экземпляров (Tao et al., 2014, 2015; Philbin et al., 2007; Tolas et al., 2015), когда заданный участок изображения-запроса ищут в пакете изображений. Использование обучения сопоставлению (Tao et al., 2015) позволяет выполнять точный поиск экземпляров объектов, даже если объект поиска представлен совершенно иначе, чем целевое изображение. При отслеживании обучение функции сопоставления полностью выполняется на примерах отслеживания. После обучения функция сопоставления способна сравнивать патчи новых экземпляров объектов или даже новых типов, которые функция раньше не встречала.

10.3.4.2. Эквивариантность при отслеживании

Сначала мы дадим краткое введение в эквивариантные модели. Затем опишем популярные виды эквивариантности, применяемые в сиамских треке-

рах. За более общим обзором эквивариантности мы отсылаем заинтересованного читателя к публикации (Weiler et al., 2018).

Свойство эквивариантности требует, чтобы функция обладала коммутативностью области определения и области значений. Для любой данной группы преобразований G функция отображения $h : X \rightarrow Y$ является эквивариантной, если она удовлетворяет условию

$$h(\rho_g^X(x)) = \rho_g^Y(h(x)) \quad g \in G, x \in X, \quad (10.23)$$

где $\rho_g^{(\cdot)}$ обозначает групповое действие в соответствующем пространстве. При инвариантности ρ_g^Y будет тождественным отображением.

В качестве наглядного примера мы рассмотрим *эквивариантность переноса* (translation equivariance), изображенную на рис. 10.4. В этом примере h обозначает функцию сверточной нейронной сети, а ψ_g обозначает группу переноса. Примеры действий из этой группы включают, например, перемещение на один пиксель влево или на один вправо либо сдвиг на несколько пикселей. Таким образом, внутри группы переноса может быть определено бесконечное количество действий. Обеспечение эквивариантности сети по отношению к переносам приводит к уменьшению сложности выборки и облегчает обобщение модели в отношении вариаций переноса.

Важно отметить, что существует несколько других преобразований, помимо переноса, которые могут быть встроены в модель для повышения робастности, если эффекты этих преобразований присутствуют в данных и задаче. В качестве примеров можно назвать повороты, отражения и изменение масштаба. Для обобщения любого из этих преобразований необходимо применить эквивариантность к соответствующей группе.

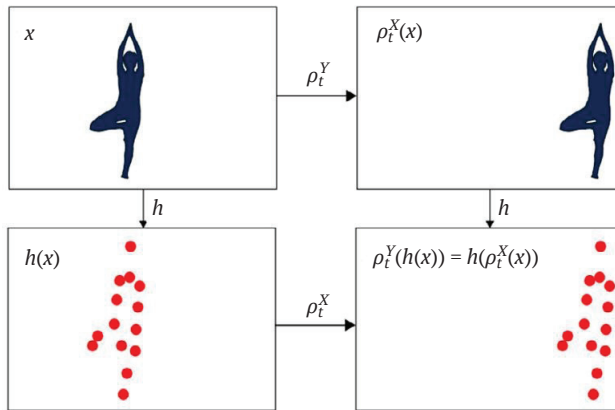


Рис. 10.4 ❖ Схематическое представление эквивариантности переноса по патчам в CNN, возникающей из-за связывания весов переноса, так что перенос ρ_t^X входного изображения x приводит к соответствующему переносу $\rho_t^Y(f(x))$ карт признаков $h(x)$. Здесь $h(\cdot)$ и $\rho_t(\cdot)$ обозначают функцию кодирования нейронной сети и функцию переноса соответственно. Источник: Worrall et al., 2017

10.3.4.3. Эквивариантность переноса

При распознавании изображений в целом и визуальном отслеживании объектов в частности варианты переноса объекта не связаны с категорией объекта, но имеют отношение к его местоположению. Модель отслеживания должна быть эквивариантна по отношению к переносу (Li et al., 2019a), чтобы она могла точно возвращать местоположение целевого объекта в любом будущем кадре.

К счастью, эквивариантность переноса почти идеально интегрирована во все модели трекеров, основанные на сверточных нейронных сетях, благодаря характеру свертки, которая является эквивариантной по отношению к оператору сдвига (Bronstein et al., 2017). На практике, однако, и особенно в контексте визуального отслеживания объектов, эквивариантность переноса не может быть идеально реализована из-за ограничивающих допущений. Как отмечено в (Li et al., 2019), сверточные нейронные сети являются эквивариантными к переносу именно тогда, когда на границах изображения нет отступов. Однако если на границах есть заполнители изображения, выходные данные свертки не будут точно равными при смещении. Это становится особенно актуальным при увеличении глубины сверточной нейронной сети, используемой для реализации сиамской функции сопоставления. Из-за увеличения действующего рецептивного поля (Simonyan, Zisserman, 2014) признаки в более глубоких слоях будут сильно зависеть от входных пикселей вблизи границ изображения, и, таким образом, модель перестает быть действительно эквивариантной к переносу. В результате модель функции сходства пространственно смещена и склонна возвращать наиболее достоверные участки прогнозов, которые находятся рядом с центром изображения. В качестве примера обратимся к рис. 10.5 (Li et al., 2019), на котором цели размещают в случайных местах, отобранных в диапазоне $(0, m)$, где 0 – центр пятна (то есть центр целевого объекта во время обучения сиамской функции подобию), а $m = 0, 16, 32$ – максимальное смещение цели в патче. При $m = 0$ цель всегда находится в центре патча, показывая, что трекер будет иметь тенденцию предсказывать центральные местоположения независимо от фактического местоположения цели.

Наличие пространственного смещения в модели трекера нежелательно, поскольку цель может находиться в любом месте кадра изображения. Возможно, что более важно, самым большим следствием этого пространственного смещения является то, что обычные сиамские трекеры не могут использовать очень глубокие сверточные нейронные сети. Тем не менее очень глубокие сверточные нейронные сети, такие как нейронные сети ResNet (He et al., 2016), неоднократно показывали, что они имеют решающее значение для повышения точности распознавания объектов.

Поскольку эквивариантность переноса не получается точно реализовать в глубоких нейронных сетях из-за несовершенных предположений, в (Li et al., 2019) предлагают практическое решение в виде *пространственно-ориентированной выборки* (spatially aware sampling). В частности, при создании обучающей выборки для обучения функции сиамского подобию предлагается добавлять шум к местоположению целевого объекта, чтобы он больше не располагался в центре. При этом пространственное смещение уменьшается

(рис. 10.5), и можно использовать более глубокие нейронные сети, такие как ResNet-50. Применение более глубоких сетей, в свою очередь, приводит к значительному повышению точности. Для получения дополнительных сведений о точной архитектуре мы отсылаем заинтересованного читателя к оригинальной публикации.

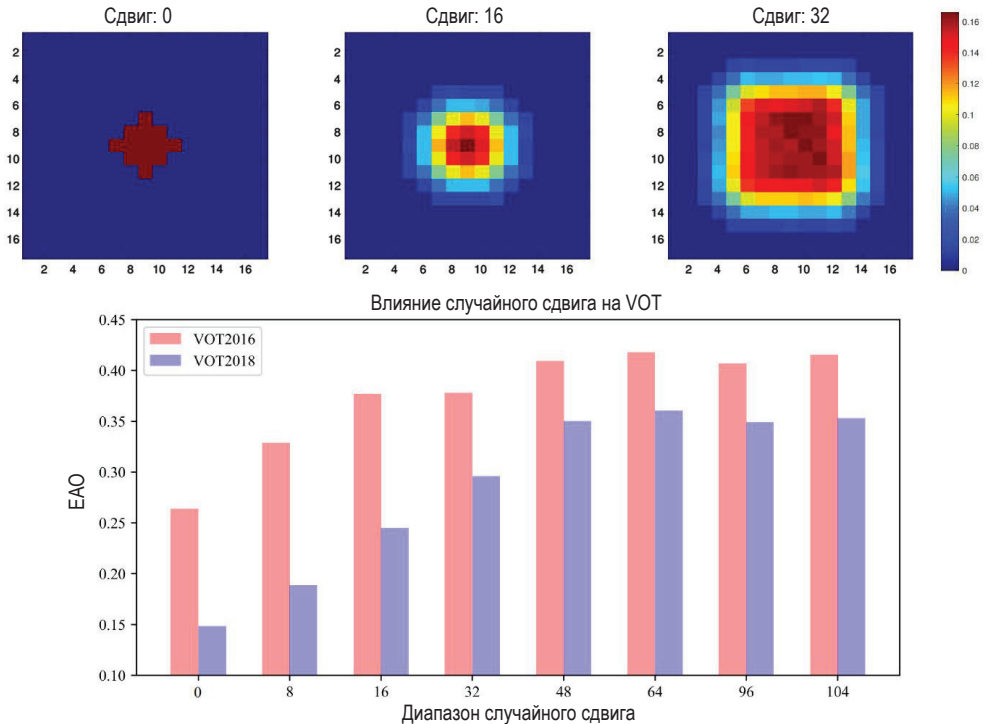


Рис. 10.5 ❖ Вверху: если не добавлять никакого случайного сдвига (сдвиг: 0) в местоположении ограничивающей рамки во время обучения, модель склонна предсказывать расположение центра, что, строго говоря, подразумевает отсутствие эквивариантности к переносу. Добавление шума к эталонной ограничивающей рамке (сдвиг: 16 или 32) помогает более равномерно распределять прогнозы. Внизу: добавление некоторого шума к эталонным предсказаниям восстанавливает строгую эквивариантность переноса, заметно улучшая точность отслеживания. Источник: Li et al., 2019

В заключение важно пояснить, почему более глубокие сети оказались успешными в классификации, но не в визуальном отслеживании объектов. В классификации изображений и объектов строгая эквивариантность переноса не обязательно требуется, потому что цель не состоит в том, чтобы предсказать точное местоположение объекта. Следовательно, пространственное смещение не приносит вреда, пока модель учится успешно распознавать категорию объекта. Напротив, при визуальном отслеживании объекта задача состоит в том, чтобы точно предсказать местоположение объекта, и пространственное смещение неуместно.

10.3.4.4. Эквивариантность вращения

Вращение является одной из наиболее распространенных, но до сих пор нерешенных сложных проблем, возникающих при визуальном отслеживании объектов. Вращение часто встречается в реальных сценариях, особенно когда камера записывает вид сверху, как в дронах, где вращается либо объект, либо сама камера. Еще одним примером являются селфи-видео, в которых объекты часто вращаются в плоскости.

Алгоритмы отслеживания, основанные на глубоком обучении, применяют глубокие сверточные нейронные сети, которые теоретически являются эквивариантными к переносу, но не предназначены для обработки вращений в плоскости. Предполагается, что модель может хорошо работать с ориентациями объекта, представленными в обучающем наборе, но не справляться с незнакомыми ориентациями, как показано на рис. 10.6.

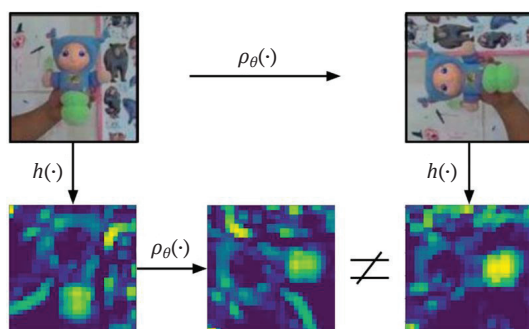


Рис. 10.6 ❖ Пример, демонстрирующий отсутствие эквивариантности вращения в обычных моделях CNN, применяемых для отслеживания объектов, $\rho_\theta(h(\cdot)) \neq h(\rho_\theta(\cdot))$. Здесь $h(\cdot)$ и $\rho_\theta(\cdot)$ обозначают функцию кодирования нейронной сети и преобразование вращения соответственно. Источник: Gupta et al., 2021

Как и в разделе 10.3.4.3, простой подход к принудительному изучению вариантов вращения заключается в использовании обучающих наборов данных, в которых повороты в плоскости происходят естественным образом или посредством дополнения данных. Однако, как подчеркивают Лаптев и др. (Laptev et al., 2016), существует несколько ограничений стратегий дополнения данных. Во-первых, такие процедуры потребуют изучения отдельных представлений для разных повернутых вариантов данных. Во-вторых, чем больше вариантов учитывается, тем более гибкой должна быть модель трекера, чтобы охватить их все. Это означает значительное увеличение обучающих данных и вычислительных расходов. Далее, такой подход сделал бы модель инвариантной к поворотам, но не эквивариантной к ним. Следовательно, прогнозы будут ненадежными, когда цель окружена похожими объектами, совершающими разные вращения, например при отслеживании рыбы в косяке других рыб.

Альтернативой является реализация эквивариантности вращения с использованием группо-эквивариантных CNN (Cohen, Welling, 2016) и управ-

ляемых фильтров (Weiler et al., 2018), чтобы сделать сиамские трекары эквивариантными к вращениям. Этот подход к эквивариантности вращения вызывает встроенное разделение весов между различными группами вращений и добавляет в модель внутреннее понятие вращения.

Управляемые фильтры. Управляемые фильтры позволяют эффективно вычислять отклики для произвольного числа оборотов дискретного фильтра L . Более того, они также демонстрируют сильные выразительные возможности. Фильтр Ψ является вращательно управляемым, если его поворот на произвольный угол θ может быть выражен через фиксированный набор атомарных функций (Freeman et al., 1991; Weiler et al., 2018). В работе (Gupta et al., 2021) определяют базис управляемых функций, используя круговые гармоники ψ_{jk} :

$$\psi_{jk}(r, \omega) = \tau_j(r)e^{ik\omega}, \quad (10.24)$$

где $\omega \in (-\pi, \pi]$ и $j = 1, 2, \dots, J$ позволяют управлять радиальной частью базисных функций. Далее, член (r, ω) относится к преобразованной версии (x_1, x_2) в полярных координатах, а $k \in \mathbb{Z}$ обозначает угловую частоту. Преимущество круговых гармоник состоит в том, что можно просто выразить повороты на ψ_{jk} как умножение на комплексную экспоненту:

$$\rho_\theta \psi_{jk}(x) = e^{-ik\theta} \psi_{jk}(x), \quad (10.25)$$

где $\theta \in (-\pi, \pi]$. Для ясности мы выражаем $\psi_{jk}(\cdot)$ как $\psi_{jk}(x)$. Затем каждый обученный фильтр строится как линейная комбинация элементарных фильтров:

$$\Psi(x) = \sum_{j=1}^J \sum_{k=0}^K w_{jk} \psi_{jk}(x) \quad (10.26)$$

с весами $w_{jk} \in \mathbb{C}$. Для поворота на θ составным фильтром можно управлять с помощью фазовой манипуляции элементарных фильтров:

$$\rho_\theta \Psi(x) = \sum_{j=1}^J \sum_{k=0}^K w_{jk} e^{-ik\theta} \psi_{jk}(x). \quad (10.27)$$

Единую ориентацию фильтра можно получить, взяв действительную часть Ψ , обозначаемую как $\text{Re } \Psi(x)$.

Сиамские трекары имеют две ветви сверточных нейронных сетей. Обе ветви в стандартных сиамских трекерах получают один вход, который является либо целевым патчем в первом кадре, либо каждым новым кадром в последовательности. Чтобы иметь вращательно-эквивариантный сиамский трекер, сверточные нейронные сети в обеих ветвях используют вращательно-эквивариантные управляемые фильтры.

Эквивариантный вход вращения. Сосредоточившись на ветви шаблонов, Гупта и др. (Gupta et al., 2021) изменяют структуру шаблона, чтобы он содержал несколько повернутых вариантов первого кадра, определенного множеством Z , где $Z = \{z_1, z_2, \dots, z_L\}$. Каждый повернутый вход I содержит C каналов, где каждый канал представлен посредством I_c и $c \in \{1, 2, \dots, C\}$. Этот

вход свернут с \hat{C} повернутыми фильтрами $\rho_\theta \Psi_{\hat{c}\hat{c}}^{(1)}$, где $\hat{c} \in \{1 \dots \hat{C}\}$. На основании уравнения (10.27) результирующие признаки, полученные до применения нелинейной активации, будут вычисляться следующим образом:

$$y_{\hat{c}}^{(1)}(x, \theta) = \text{Re} \sum_{c=1}^C \sum_{j=1}^J \sum_{k=0}^{K_j} w_{\hat{c}cj k} e^{-ik\theta} (I_c * \psi_{jk})(x), \quad (10.28)$$

где фильтры представляют собой повернутые варианты с равноудаленными ориентациями θ , представленными множеством $\theta = \left\{0, \Lambda, \dots, 2\pi \frac{\Lambda-1}{\Lambda}\right\}$. Затем применяются члены смещения $\beta_{\hat{c}}^{(1)}$ и нелинейность σ , чтобы получить карту признаков на первом слое $\zeta_{\hat{c}}^{(1)}$.

Обратите внимание, что необходимо повернуть входные данные только на одной из двух ветвей, так как вращения в свертках в уравнении (10.28) относительно. Имеет смысл выполнять поворот ввода в ветке шаблона, так как это можно сделать один раз на первом кадре.

Также отметим, что теоретически, вместо того чтобы брать все возможные версии вращения Z шаблонной цели, также можно сначала вычислить функцию $h(z)$ исходной цели, а затем повернуть $h(z)$. Однако на практике пространственное разрешение $h(z)$ очень низкое, обычно 6×6 или 7×7 пикселей. В результате из-за грубого преобразования возникнут артефакты по углам и краям. Лучше сначала повернуть весь кадр (а не только цель) вокруг центра вращения цели, затем обрезать и вычислить $h(z)$. Поскольку упомянутые операции выполняются только в ветке цели, этот шаг может быть рассчитан заранее и внесет лишь незначительные дополнительные затраты.

Вращательно-эквивариантные свертки. Карты признаков, полученные из уравнения (10.28), обрабатываются далее с использованием групповых сверток, обобщающих пространственные свертки на более широкое множество групп преобразований. Подобно первому слою, управляемые фильтры определяются для группы как

$$y_{\hat{c}}^{(l)}(x, \theta) = \text{Re} \sum_{c=1}^C \sum_{\omega \in \Omega} \sum_{j,k} w_{\hat{c}cj k, \theta - \phi} e^{-ik\theta} (h_c^{(l-1)}(\cdot, \omega) * \psi_{jk})(x), \quad (10.29)$$

где $h_c^{(l-1)}$ обозначает c -й канал карты признаков на $(l-1)$ -м слое, который теперь заменяется на h . Дополнительный индекс $\theta - \omega$, введенный в уравнении (10.29) для тензора весов, облегчает операцию групповой свертки по измерению вращения. Он подразумевает преобразование функций в группе путем их пространственного вращения.

Вращательно-эквивариантный пулинг. Выходные данные последнего сверточного слоя группы дополнительно обрабатываются путем пулинга по измерению вращения. В отличие от обычных задач классификации, пулинг не выполняется по пространственному измерению. Причина в том, что мы хотим сохранить эквивариантность вращения.

Вращательно-эквивариантные сямские трекары. Из двух ветвей мы получаем два множества карт признаков, $\{h(z)\}$ и $h(x)$, где $\{h(z)\}$ – множество,

содержащее карты признаков в Λ ориентациях. Затем $\{h(z)\}$ и $h(x)$ свертываются для получения $\{f(z, x)\}$ – множества из Λ карт интенсивности (heatmap), где $f_i(z, x) = h(z_i) * h(x)$ для всех $z_i \in Z$. Далее $\{f(z, x)\}$ обрабатывается с помощью глобальной операции тах-пулинга для получения конечной выходной карты интенсивностей $f(Z, x)$, где Z – множество значений z для нескольких ориентаций шаблона. Глобальная операция тах-пулинга находит максимальное значение в $\{f(z, x)\}$ и выбирает карту признаков, которая его содержит.

Вышеупомянутые модули представляют собой необходимый набор компонентов для сиамского трекера, эквивариантного вращению. Все сверточные слои и слои пулинга должны быть заменены их аналогами, эквивариантными повороту. Сначала необходимо определить точность трекера с точки зрения различения различных ориентаций в соответствии с вращательными степенями свободы. При наличии Λ групп вращения трекер был бы совершенно эквивариантен углам, определяемым множеством $\Omega = \{(i - 1) \cdot 360/\Lambda\}_{i=1}^{\Lambda}$. Для генерации $h(z, x)$ выполняются Λ групповых сверток с целью получения Λ различных карт интенсивности. Наконец, выполняется глобальный тах-пулинг карт признаков для генерации $h(Z, x)$, который затем обрабатывается для локализации цели.

Окончательный трекер эквивариантен только вращению в плоскости, поскольку вращение вне плоскости требует знания 3D-сцены. Дополнительным побочным преимуществом сиамских трекеров, эквивариантных вращению, по сравнению с моделями, инвариантными к вращению, является то, что их можно использовать для вычисления изменений относительной ориентации цели путем обучения без учителя. Более того, их можно использовать для добавления дополнительных регуляризаций и ограничений для обеспечения последовательности вращательного движения. Для получения дополнительной информации об устройстве сиамских трекеров, эквивариантных вращению, мы отсылаем заинтересованного читателя к (Gupta et al., 2021).

10.3.4.5. Эквивариантность масштаба

Другой тип изменений, который часто встречается при отслеживании видеообъектов, – это вариации масштаба. Точное измерение масштаба имеет решающее значение, когда камера использует увеличивающий объектив или когда цель перемещается по глубине. Как отмечают Сосновик и др. (Sosnovik et al., 2021), масштаб также важен для дифференциации различных объектов при отслеживании, особенно когда многие объекты на видео имеют схожий внешний вид, например во время трансляции спортивного матча или съемок толпы. В таких обстоятельствах эквивариантность в пространственном масштабе обеспечивает более различительное представление, которое необходимо для надежной дифференциации нескольких похожих кандидатов. Важно отметить, что масштабно-пространственная эквивариантность помогает поддерживать лучшую стабильность размера предсказанных ограничивающих рамок, даже если изменения масштаба в видео незначительны.

Обычный способ реализации масштаба в трекере – это обучение сети на большом наборе данных, где изменения масштаба происходят естественным образом. Как отмечают Лаптев и др. (Laptev et al., 2016) и было сказано в пре-

дыдущем подразделе, такие процедуры обучения могут привести к появлению масштабированных дубликатов почти одинаковых фильтров, что делает оценку межмасштабного сходства ненадежной. Масштабно-эквивариантные модели имеют внутреннее понятие масштаба и встроенное распределение веса между различными масштабами фильтров. Таким образом, эквивариантность масштаба направлена на получение одинакового представления для всех размеров.

Масштабно-эквивариантные свертки. Для заданной функции $\rho_s : \mathbb{R} \rightarrow \mathbb{R}$ масштабное преобразование определяется следующим образом:

$$L_s[f](t) = f(\rho_s^{-1}t), \forall s \geq 0, \quad (10.30)$$

где случаи с $\rho_s > 1$ называются *укрупняющим масштабированием*, или *апскейлингом* (upscaling), а с $\rho_s < 1$ – *уменьшающим масштабированием*, или *даунскейлингом* (downscaling). Стандартные сверточные слои и сверточные сети являются трансляционно-эквивариантными, но не масштабнo-эквивариантными (Sosnovik et al., 2020).

Чтобы построить масштабнo-эквивариантные сверточные сети, Сосновик и др. (Sosnovik et al., 2020) начинают с выбора полного базиса функций следующего вида, определенных в нескольких масштабах, выбирая центром функции точку $(0, 0)$ в координатах (u, v) :

$$\psi_{\sigma nm}(u, v) = A \frac{1}{\sigma^2} H_n\left(\frac{u}{\sigma}\right) H_m\left(\frac{v}{\sigma}\right) e^{-\frac{u^2 + v^2}{2\sigma^2}}. \quad (10.31)$$

В отличие от управляемых фильтров, здесь H_n – полином Эрмита n -го порядка и A – константа, применяемая для нормализации. Построение базиса из N функций возможно путем итерации по возрастающим парам n и m . Для полного и фиксированного базиса количество функций N равно количеству пикселей в исходном фильтре с выбранным множеством эквидистантных масштабов σ :

$$\Psi_\sigma = \{\psi_{\sigma 00}, \psi_{\sigma 01}, \psi_{\sigma 10}, \psi_{\sigma 11}, \dots\}. \quad (10.32)$$

Наконец, свертки изучаются как взвешенные комбинации $\psi_{\sigma i}$ с использованием обучаемых весов w :

$$\kappa_\sigma = \sum_i \Psi_{\sigma i} w_i. \quad (10.33)$$

В результате окончательные ядра определяются в нескольких масштабах, и интерполяция изображения не требуется. Для заданной функции масштаба и переноса $f(s, t)$ масштабная свертка с ядром $\kappa_\sigma(s, t)$ равна

$$[f \star_H \kappa_\sigma](s, t) = \sum_{s'} [f(s', \cdot) \star \kappa_{s \cdot \sigma}(s^{-1}s', \cdot)](t), \quad (10.34)$$

где \star_H обозначает масштабнo-эквивариантную свертку. Результатом этой свертки является набор признаков, каждый из которых соответствует разным масштабам.

Эквивариантные пулинг и паддинг. Чтобы захватить корреляции между различными масштабами и преобразовать трехмерный сигнал в двумерный, необходимо применить глобальный тах-пулинг вдоль оси масштаба. Эта операция не исключает масштабную эквивариантность сети. На практике Сосновик и др. (2021) рекомендуют дополнительно включать масштабно-эквивариантный пулинг в местах, где обычные CNN содержат пространственный тах-пулинг или страйды.

Показано, что паддинг снижает способность сверточных трекеров к локализации объектов (Li et al., 2019; Zhang, Peng, 2019), как было сказано также в разделе 10.3.4.3. Однако масштабные эквивариантные свертки полагаются на большие рецептивные поля, которые, следовательно, дают меньшие карты признаков. Следовательно, для масштабно-эквивариантных трекеров с очень глубокими моделями необходим паддинг. Чтобы гарантировать, что строгая эквивариантность переноса не будет нарушена при одновременном получении благоприятных результатов, в (Sosnovik et al., 2021) используют циклический паддинг во время обучения и нулевой – во время вывода.

Масштабно-эквивариантные сиамские трекеры. Операция свертки, в результате которой получается карта интенсивностей трекера, непараметрична. И входные данные, и ядро поступают из нейронных сетей. Таким образом, подход, реализованный в уравнении (10.34), для этого случая не подходит. Для двух функций f_1, f_2 масштаба и переноса непараметрическая масштабная свертка определяется следующим образом:

$$[f_1 \star_H f_2](s, t) = L_{s^{-1}}[L_s[f_1] \star f_2](t). \quad (10.35)$$

Здесь L_s – масштабирование, реализованное как бикубическая интерполяция. Хотя это относительно медленная операция, она используется в трекере только один раз и не сильно влияет на время вывода. Вышеупомянутые модули являются необходимыми и достаточными компонентами для построения масштабно-эквивариантного сиамского трекера. А именно необходимо сначала определить, в какой степени объекты изменяются в размерах в этой области, а затем соответственно выбрать набор масштабов $\sigma_1, \sigma_2, \dots, \sigma_N$. Это гиперпараметр, специфичный для предметной области, в которой работает сеть. Для сетей, представленных $h(x)$ и $h(z)$, все сверточные слои необходимо заменить масштабно-сверточными. Базисом этих слоев служат выбранные масштабы $\sigma_1, \sigma_2, \dots, \sigma_N$. При желании можно включить пулинг по масштабу, чтобы дополнительно извлечь межмасштабные корреляции между всеми масштабами. Параметрическая масштабно-эквивариантная свертка должна быть заменена непараметрической из уравнения (10.35), чтобы получить окончательную карту прогноза.

Полученный трекер создает карту интенсивностей $f(z, x)$, определенную для масштаба и переноса. Каждой позиции назначается вектор признаков, который кодирует как меру подобия, так и отношение масштаба между кандидатом и шаблоном. Если имеется дополнительный пулинг масштаба, то вся информация о масштабе агрегируется в показателе подобия. Поскольку не меняются ни общая структура трекера, ни процедуры обучения и вывода, дополнительные вычислительные затраты за счет введения масштабной эквивариантности невелики. За дополнительной информацией об устройстве

сиамских трекеров, эквивалентных масштабу, мы отправляем заинтересованного читателя к работе (Sosnovik et al., 2021).

10.3.4.6. Эффективность сиамских трекеров

Первый сиамский трекер от (Taо et al., 2016) был основан на архитектуре быстрой RCNN (Girshick, 2015) и пулинге области наблюдения (region-of-interest, ROI) для выполнения сопоставления с локальным шаблоном. Несмотря на повышение скорости, обеспечиваемое процедурой пулинга ROI, поиск по рамкам по-прежнему является процедурой, требующей значительных вычислительных ресурсов. Бертинетто и др. (Bertinetto et al., 2016) отмечают, что поиск по сверточным картам признаков сам может быть описан как свертка, исходя из принципа работы полностью сверточных нейронных сетей (Long et al., 2015). Это простое изменение сделало сиамские трекеры намного более эффективными и сравнимыми с предыдущими альтернативами.

Поскольку поиск по всем возможным местам на изображении обходится дорого, в (Li et al., 2018) предлагают ввести в свой сиамский трекер *сеть предсказания области* (region proposal network) от (Ren et al. (2015)). Сеть предсказания области учится регрессии по координатам местоположения и масштабу рамки, которая, вероятно, содержит цель. Поэтому в своем поиске сиамский трекер может ориентироваться только на нужные регионы. Чтобы еще больше сократить объем вычислений, в (Li et al., 2019) предлагают свертки по глубине, а не обычные свертки, получая в десять раз меньше параметров и значительное преимущество в вычислительном отношении.

10.3.4.7. Гибридное обучение с сиамскими трекерами

Сиамские трекеры продемонстрировали большие преимущества в точном прогнозировании местоположения целей, особенно в длинных последовательностях, благодаря тому, что они не страдают от устаревания модели. Это возможно, если полагаться исключительно на автономное обучение и не включать какой-либо компонент последовательного обучения в процессе работы. Однако игнорирование всех будущих появлений цели и отсутствие последовательного обучения противоречат здравому смыслу. Последовательное обучение необходимо для работы со сценариями, в которых внешний вид цели значительно изменяется в ходе отслеживания, так что полагаться только на первый кадр недостаточно. Например, представьте себе случай, когда пешеход снимает куртку. Последовательная составляющая также имеет решающее значение в случае, если в кадре присутствует несколько похожих объектов и модель должна их различать, что является сложной задачей для простых сиамских трекеров. В этом случае последовательное обучение помогает модели трекера отличить целевой объект от всех других подобных объектов на сцене.

Чтобы убедиться, что последовательное обучение не оказывает пагубного влияния на модель трекера из-за устаревания модели, обновления необходимо применять с осторожностью. При разработке механизмов обновления модели следует учитывать два аспекта. Первый аспект заключается в том,

когда и как часто надо выполнять обновление модели. Чем чаще обновляются модели, тем больше вероятность того, что модель трекера будет становиться все более и более предвзятой со временем, как показано в (Gavves et al., 2020). Второй аспект заключается в том, какую именно часть модели следует обновить, чтобы гарантировать, что возникшее смещение будет либо небольшим, либо окажет минимальное влияние в долгосрочной перспективе на сопоставление с шаблоном. Обновления могут быть введены либо непосредственно в функции подобию сиамского трекера (Bhat et al., 2019), либо во вспомогательные сети, дополняющие функцию подобию (Tao et al., 2017).

Чтобы ввести компонент последовательного обучения, многие подходы предлагают стратегию *метаобучения*. Согласно этой стратегии модель трекера изначально обучается в автономном режиме, чтобы оптимизировать веса параметров на просмотренных данных. Затем во время вывода мета-обучаемая модель предсказывает новые веса параметров, оптимальные для новой точки данных.

Бхат и др. (Bhat et al., 2019) предлагают компонент метаобучения, который прогнозирует новые веса параметров, различающиеся для цели и фона. Что касается функции сходства, предлагаемая модель основана на стандартной сиамской архитектуре. Однако, в отличие от стандартных сиамских трекеров, за последним сверточным слоем в ветке шаблона следуют *инициализатор модели* и модуль *оптимизатора модели*. Инициализатор модели обеспечивает начальное вычисление весов модели, используя только внешний вид цели. Эти веса затем обрабатываются модулем оптимизатора модели, который использует как цель, так и фон для получения окончательных параметров веса. Чтобы избежать любого потенциального переобучения, оптимизатор модели содержит очень мало обучаемых параметров. В итоге модель обучается как автономно, так и последовательно в процессе работы.

На автономном этапе модель обучается с использованием пар наборов ($M_{\text{train}}, M_{\text{test}}$). Каждый набор содержит N кадров, $M = \{I_j, b_j\}$, где b_j – целевые ограничивающие рамки, доступные во время обучения. Чтобы построить M_{train} и M_{test} , для каждой обучающей последовательности выбирают случайный сегмент, а затем делят его на две части. Затем предсказатель модели получает в качестве входных данных сверточные карты признаков, вычисленные из M_{train} , на основании которых предсказывает веса параметров. Потом, чтобы обеспечить хорошее обобщение, эти веса параметров используются для прогнозирования отслеживания только на M_{test} . На этапе последовательного обучения первоначальный внешний вид цели искажается с использованием методов увеличения данных для создания новых образцов обучающей выборки. Этот набор дополняется новыми патчами цели всякий раз, когда цель прогнозируется с достаточной уверенностью. Последовательное обучение по своей сути аналогично автономному обучению, но выполняется через равные промежутки времени каждые 20 кадров.

Двигаясь в том же направлении, в работе (Danelljan et al., 2020) предлагают включить вероятностное обучение в сеть, предложенную в (Bhat et al., 2019). Чтобы достичь этого, модель учится минимизировать расхождение Кульбака–Лейблера между своим выходом $p_{\theta}(y|x)$ и эталонным распределением $p(y|y_i)$. Вероятностный результат модели определяется как энергетическое

распределение $p_{\theta}(y|x) = \frac{1}{Z_{\theta}} \exp(s_{\theta}(y, x))$, где $Z_{\theta} = \int_y \exp(s_{\theta}(y, x))$. Эталонное распределение $p(y|y_i)$ оценивается эмпирически как гауссово распределение $p(y|y_i)\mathcal{N}(y_i, \sigma^2)$, где σ^2 – эмпирическая дисперсия, оцениваемая по небольшой выборке данных.

В отличие от вышеупомянутых работ, (Tao et al., 2017; Gavves et al., 2020) иначе подходят к проблеме, сосредоточившись на том, когда трекер должен выполнять обновление. Идея состоит в том, что хотя сиамские трекеры могут извлечь выгоду из обновлений модели, они должны быть осторожны, чтобы не внести необратимые искажения в функцию подобия. Как видно из (10.14), точная оценка смещения модели потребует вычисления члена $E[\delta_i \nabla_{\phi} f_{i,t}]$. Однако параметры состояния ϕ_{t+1} заранее неизвестны, и повторное обнаружение цели в более ранних кадрах для каждого шага обновления модели (для получения $f_{i,t}$) было бы слишком сложным с вычислительной точки зрения. В связи с этим было предложено использовать каскадную нейронную сеть для определения необходимости выполнения обновления модели. Сначала сиамский трекер используется для вычисления карты подобия после свертки карты признаков шаблона с картой признаков из каждого нового кадра. За функцией сиамского подобия следует *сеть распознавания устаревания* (decay recognition network), реализованная в виде бинарного классификатора на основе LSTM. Классификатор LSTM получает в качестве входных данных предыдущие K карт подобия, возвращенных сиамским трекером. Чтобы убедиться в отсутствии предвзятости LSTM, его также обучают в автономном режиме. Когда классификатор LSTM возвращает положительный прогноз, веса модели обновляются. Вдобавок, вместо того чтобы полагаться только на предсказание области для ускорения поиска, модель выполняет глобальный поиск каждые T кадров, так что сиамский трекер не теряет цель, если она оказывается за пределами области поиска. Важно подчеркнуть, что в этой модели глобальный поиск принудительно выполняется через равные промежутки времени, чтобы отделить обновления модели от прогнозов. Если бы обновления зависели от прогнозов модели, это создало бы зависимость обновлений от поведения модели, что привело бы к самообусловленным обновлениям и в конечном итоге к разрушению модели.

Аналогичным образом Дай и др. (Dai et al., 2020) предлагают обучаемый в автономном режиме *метаобновитель* (metaupdater), который призван решить, следует ли обновлять модель трекера. Метаобновитель опирается на последовательность факторов, состоящую из (а) геометрии ограничивающей рамки, (б) оценки достоверности и (в) изменения внешнего вида с течением времени. Во время обучения трекер полагается на метаобновитель, чтобы решить, следует обновлять параметры веса или нет. Затем параметры метаобновителя оптимизируются, чтобы делать правильные прогнозы отслеживания в будущих последовательностях.

Разработка гибридных моделей трекеров, которые обучаются как в автономном, так и в последовательном режимах, при этом *принципиально исключая* или сводя к минимуму распад модели, все еще остается открытым вопросом и предметом исследований.

10.3.4.8. Последовательное обучение помимо сиамских трекеров

До сиамских трекеров также было предпринято несколько важных попыток моделирования трекеров в длинных видеопоследовательностях. В своей основополагающей работе Калал и др. (Kalal et al., 2012) предлагают разделить длительное отслеживание на задачи моделирования отслеживания, обучения и обнаружения. Трекер отвечает за оценку движения объекта от одного кадра к другому. Допущение компонента трекера заключается в том, что движение между кадрами ограничено и объект всегда виден. Детектор обрабатывает каждый кадр независимо, чтобы исправить любые ошибки, допущенные трекером. Ложноположительные и ложноотрицательные результаты, возвращаемые трекером и предсказателем, затем отслеживаются и корректируются обучающим компонентом. Обучение опирается на пару экспертов. Р-эксперт выявляет только ложноотрицательные результаты, а N-эксперт – только ложноположительные. Поскольку оба эксперта сами могут ошибаться, они остаются независимыми, чтобы была гарантия, что их индивидуальные ошибки взаимно компенсируются. Предлагаемый трекер особенно хорошо подходит для долгосрочного отслеживания благодаря механизму самокоррекции, используемому независимым детектором и компонентами обучения.

В статье (Pernici, Del Bimbo, 2013) также предложен трекер, который хорошо подходит для длительного отслеживания. С этой целью трекер использует передискретизацию локальных инвариантных представлений SIFT (Lowe, 2004), которые используются в качестве обучающих выборок, передаваемых паре дискриминативных классификаторов ближайших соседей. Классификатор целевого объекта пытается смоделировать внешний вид целевого объекта, сравнивая внешний вид и форму с другими соседними патчами. Классификатор контекста пытается смоделировать внешний вид пространственно-временного фона. Кроме того, трекер использует случайный поиск, когда объект отсутствует в течение определенного количества последовательных кадров. Важным компонентом трекера является геометрическое сопоставление, основанное на голосовании в стиле RANSAC. Когда количество совпадений меньше ожидаемого, объект предположительно заслоняют другие объекты. Модель обновляется при каждом успешном обнаружении, если нет окклюзии.

По сравнению с сиамскими трекерами, эти более ранние методы не изучали функцию подобию в автономном режиме и должны были полагаться на сложные механизмы, чтобы убедиться, что дрейф модели ограничен. Тем не менее тип сигналов, используемых в этих методах, имеет определенное сходство с сиамскими трекерами. Опираясь на сверточные нейронные архитектуры, подобные (Tao et al., 2016; Bertinetto et al., 2016), они учитывают как внешний вид, так и слабые геометрические признаки при определении сходства целевого объекта с возможными местоположениями в новых кадрах, как это делают (Kalal et al., 2012; Pernici, Del Bimbo, 2013). Более того, в трекерах (Tao et al., 2017; Kalal et al., 2012) используют поиск по полному изображению, чтобы свести к минимуму ложноотрицательные результаты.

10.3.5. Наборы данных и тесты

Различные бенчмарки отслеживания (Smeulders et al., 2014; Wu et al., 2015; Kristan et al., 2016; Liang et al., 2015; Li et al., 2016; Mueller et al., 2016; Valmadre et al., 2018) сыграли огромную роль в развитии этой области, позволив провести объективное сравнение различных методов и добившись впечатляющего прогресса в последние годы. Эти бенчмарки предназначены для тестирования краткосрочного отслеживания в соответствии с определением (Kristan et al., 2016), которое не подразумевает наличия методов для повторного обнаружения. Это означает, что объект всегда присутствует в видеокадре. Однако существующие бенчмарки также краткосрочны в буквальном смысле, так как средняя продолжительность видео не превышает 20–30 секунд. Короткая продолжительность видео имеет несколько последствий для оценки алгоритмов долгосрочного отслеживания.

Одним из следствий этого является то, что при использовании коротких видеороликов для отслеживания неблагоприятные последствия устаревания модели трудно заметить просто потому, что выполняется недостаточное количество обновлений модели. Даже на умеренно длинных видеопоследовательностях краткосрочные трекеры чаще всего становятся настолько смещенными, что полностью упускают цель и, что, возможно, более важно, не могут восстановиться (Gavves et al., 2020). Сверх того, поскольку целевые объекты редко покидают кадр, невозможно оценить, хорошо ли алгоритмы справляются со случаями исчезновения и повторного появления объектов. Наконец, в случае более длинных видеороликов критерии успеха длительного отслеживания могут отличаться от критериев для более коротких последовательностей. Например, точная попиксельная локализация целевого объекта в коротких видеороликах может дать очень высокие оценки по некоторым показателям. Однако эта попиксельная локализация становится менее важной, если вскоре после этого трекер полностью теряет цель.

По этой причине для длительного отслеживания нужны новые наборы данных, ориентиры и типы оценок (Fan et al., 2019; Valmadre et al., 2018; Mueller et al., 2016; Huang et al., 2019; Mueller et al., 2018), где видео обычно намного длиннее, в диапазоне от нескольких минут до получаса, а целевые объекты часто появляются и исчезают.

10.4. ЗАКЛЮЧЕНИЕ

Визуальное отслеживание объектов – одна из старейших задач компьютерного зрения. Устаревание модели в длинных последовательностях, исчезновение и повторное появление цели создают серьезные проблемы для краткосрочных моделей отслеживания. Успех глубокого обучения повлиял на визуальное отслеживание объектов, особенно в контексте долгосрочных последовательностей. Причина в том, что благодаря сиамской архитектуре глубокого трекера можно перевести все сравнения внешнего вида в автономное изучение функции подобию и избежать устаревания модели. Хотя сиамские трекеры невосприимчивы к устареванию модели, они страдают от потери

объекта в тех случаях, когда внешний вид цели значительно меняется по сравнению с первым кадром. Для решения этой проблемы предложены сиамские трекары со встроенными инвариантностями и эквивариантностями, а также гибридные сиамские трекары, которые полагаются как на автономное, так и на последовательное обучение. Эти варианты сиамских трекаров могут учитывать большие различия во внешнем виде целевого объекта, но при этом демонстрируют небольшое устаревание модели. Исходя из данных факторов можно сказать, что сиамские трекары хорошо себя зарекомендовали и рекомендуются для длительного визуального отслеживания объектов.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Baker S., Matthews I.*, 2004. Lucas-Kanade 20 years on: a unifying framework. *International Journal of Computer Vision*.
- Bertinetto L., Henriques J. a. F., Valmadre J., Torr P. H. S., Vedaldi A.*, 2016a. Learning feed-forward one-shot learners. In: *Advances in Neural Information Processing Systems*.
- Bertinetto L., Valmadre J., Henriques J. F., Vedaldi A., Torr P. H. S.*, 2016b. Fully-convolutional Siamese networks for object tracking. In: *European Conference on Computer Vision Workshops*.
- Bhat G., Danelljan M., Gool L. V., Timofte R.*, 2019. Learning discriminative model prediction for tracking. In: *IEEE International Conference on Computer Vision*, pp. 6182–6191.
- Briechele K., Hanebeck U. D.*, 2001. Template matching using fast normalized cross correlation. In: *Optical Pattern Recognition XII, International Society for Optics and Photonics*, pp. 95–102.
- Bronstein M. M., Bruna J., LeCun Y., Szlam A., Vandergheynst P.*, 2017. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*.
- Cohen T., Welling M.*, 2016. Group equivariant convolutional networks. In: *International Conference on Machine Learning*, pp. 2990–2999.
- Comaniciu D., Ramesh V., Meer P.*, 2000. Real-time tracking of non-rigid objects using mean shift. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Dai K., Zhang Y., Wang D., Li J., Lu H., Yang X.*, 2020. High-performance long-term tracking with meta-updater. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Danelljan M., Bhat G., Shahbaz Khan F., Felsberg M.*, 2017. ECO: efficient convolution operators for tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Danelljan M., Hager G., Shahbaz Khan F., Felsberg M.*, 2015. Learning spatially regularized correlation filters for visual tracking. In: *IEEE International Conference on Computer Vision*, pp. 4310–4318.
- Danelljan M., Van Gool L., Timofte R.*, 2020. Probabilistic regression for visual tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*.
- Du D., Qi H., Li W., Wen L., Huang Q., Lyu S.*, 2016. Online deformable object tracking based on structure-aware hyper-graph. *IEEE Transactions on Image Processing* 25, 3572–3584.

- Fan H., Lin L., Yang F., Chu P., Deng G., Yu S., Bai H., Xu Y., Liao C., Ling H.*, 2019. Lasot: a high-quality benchmark for large-scale single object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5374–5383.
- Fan H., Ling H.*, 2017. Sanet: structure-aware network for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 42–49.
- Fiaz M., Mahmood A., Javed S., Jung S. K.*, 2019. Handcrafted and deep trackers: recent visual object tracking approaches and trends. *ACM Computing Surveys (CSUR)* 52, 1–44.
- Freeman W. T., Adelson E. H., et al.*, 1991. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 891–906.
- Gavves E., Gupta D., Tao R., Smeulders A.*, 2020. Model decay in long-term tracking. In: IEEE International Conference on Pattern Recognition.
- Girshick R.*, 2015. Fast R-CNN. In: IEEE International Conference on Computer Vision.
- Godec M., Roth P. M., Bischof H.*, 2013. Hough-based tracking of non-rigid objects. *Computer Vision and Image Understanding* 117, 1245–1256.
- Gupta D., Arya D., Gavves E.*, 2021. Rotation equivariant Siamese networks for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition.
- He K., Zhang X., Ren S., Sun J.*, 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Held D., Thrun S., Savarese S.*, 2016. Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision. Springer, pp. 749–765.
- Huang L., Zhao X., Huang K.*, 2019. Got-10k: a large high-diversity benchmark for generic object tracking in the wild.
- Kalal Z., Matas J., Mikolajczyk K.*, 2010. Pn learning: bootstrapping binary classifiers by structural constraints. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 49–56.
- Kalal Z., Mikolajczyk K., Matas J.*, 2012. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Čehovin L., Vojíř T., Hager G., Lukežić A., Eldesokey A., Fernandez G.*, 2017. The visual object tracking VOT2017 challenge results. In: IEEE International Conference on Computer Vision Workshops.
- Kristan M., Leonardis A., Matas J., Felsberg M., Pflugfelder R., Kamarainen J. K., Čehovin Zajc L., Danelljan M., Lukežić A., Drbohlav O., He L., Zhang Y., Yan S., Yang J., Fernandez G., et al.*, 2020. The eighth visual object tracking vot2020 challenge results. In: ECCV workshops.
- Kristan M., Matas J., Leonardis A., Vojíř T., Pflugfelder R., Fernandez G., Nebehay G., Porikli F., Čehovin L.*, 2016. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Kwon J., Lee K. M.*, 2011. Tracking by sampling trackers. In: IEEE International Conference on Computer Vision. IEEE, pp. 1195–1202.
- Kwon J., Lee K. M., Park F. C.*, 2009. Visual tracking via geometric particle filtering on the affine group with optimal importance functions. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 991–998.

- Laptev D., Savinov N., Buhmann J. M., Pollefeys M., 2016. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 289–297.
- Li A., Lin M., Wu Y., Yang M. H., Yan S., 2016. NUS-PRO: a new visual tracking challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Li B., Yan J., Wu W., Zhu Z., Hu X., 2018a. High performance visual tracking with Siamese region proposal network. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Li F., Tian C., Zuo W., Zhang L., Yang M. H., 2018b. Learning spatial-temporal regularized correlation filters for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4904–4913.
- Li B., Wu W., Wang Q., Zhang F., Xing J., Yan J., 2019a. Siamrpn++: evolution of Siamese visual tracking with very deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4282–4291.
- Li C., Liang X., Lu Y., Zhao N., Tang J., 2019b. Rgb-t object tracking: benchmark and baseline. Pattern Recognition 96, 106977.
- Liang P., Blasch E., Ling H., 2015. Encoding color information for visual tracking: algorithms and benchmark. IEEE Transactions on Image Processing.
- Long J., Shelhamer E., Darrell T., 2015. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Lowe D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision.
- Lucas B. D., Kanade T., 1981. An iterative image registration technique with an application to stereo vision. In: International Joint Conferences on Artificial Intelligence.
- Ma C., Huang J. B., Yang X., Yang M. H., 2015. Hierarchical convolutional features for visual tracking. In: IEEE International Conference on Computer Vision, pp. 3074–3082.
- Mueller M., Bibi A., Giancola S., Alsubaihi S., Ghanem B., 2018. Trackingnet: a large-scale dataset and benchmark for object tracking in the wild. In: European Conference on Computer Vision, pp. 300–317.
- Mueller M., Smith N., Ghanem B., 2016. A benchmark and simulator for uav tracking. In: European Conference on Computer Vision.
- Nam H., Baek M., Han B., 2016. Modeling and propagating cnns in a tree structure for visual tracking. arXiv preprint. arXiv:1608.07242.
- Nguyen H. T., Smeulders A. W., 2004. Fast occluded object tracking by a robust appearance filter. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Nguyen H. T., Smeulders A. W., 2006. Robust tracking using foreground-background texture discrimination. International Journal of Computer Vision 69, 277–293.
- Oron S., Bar-Hillel A., Levi D., Avidan S., 2015. Locally orderless tracking. International Journal of Computer Vision 111, 213–228.
- Pan J., Hu B., 2007. Robust occlusion handling in object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Pernici F., Del Bimbo A., 2013. Object tracking by oversampling local features, pp. 2538–2551.
- Philbin J., Chum O., Isard M., Sivic J., Zisserman A., 2007. Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition.

- Qi Y., Zhang S., Qin L., Yao H., Huang Q., Lim J., Yang M. H., 2016. Hedged deep tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4303–4311.
- Ren S., He K., Girshick R. B., Sun J., 2015. Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems.
- Ross D. A., Lim J., Lin R. S., Yang M. H., 2008. Incremental learning for robust visual tracking. International Journal of Computer Vision.
- Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C., Fei-Fei L., 2015. ImageNet large scale visual recognition challenge. International Journal of Computer Vision.
- Simonyan K., Zisserman A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.
- Smeulders A. W. M., Chu D. M., Cucchiara R., Calderara S., Dehghan A., Shah M., 2014. Visual tracking: an experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Sosnovik I., Moskalev A., Smeulders A., 2021. Scale equivariance improves Siamese tracking.
- Sosnovik I., Szmajam M., Smeulders A., 2020. Scale-equivariant steerable networks.
- Tao R., Gavves E., Smeulders A. W. M., 2016. Siamese instance search for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Tao R., Gavves E., Smeulders A. W. M., 2017. Tracking for half an hour. arXiv preprint. arXiv:1711.10217.
- Tao R., Gavves E., Snoek C., Smeulders A., 2014. Locality in generic instance search from one example. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Tao R., Smeulders A. W. M., Chang S. F., 2015. Attributes and categories for generic instance search from one example. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Tolias G., Avrithis Y., Jégou H., 2015. Image search with selective match kernels: aggregation across single and multiple images. International Journal of Computer Vision.
- Valmadre J., Bertinetto L., Henriques J., Tao R., Vedaldi A., Smeulders A., Torr P., Gavves E., 2018. Long-term tracking in the wild: A benchmark. In: European Conference on Computer Vision.
- Wang G., Wang J., Tang W., Yu N., 2017. Robust visual tracking with deep feature fusion. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp. 1917–1921.
- Weiler M., Hamprecht F. A., Storath M., 2018. Learning steerable filters for rotation equivariant cnns. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 849–858.
- Worrall D. E., Garbin S. J., Turmukhambetov D., Brostow G. J., 2017. Harmonic networks: deep translation and rotation equivariance. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5028–5037.
- Wu Y., Lim J., Yang M. H., 2015. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence.

- Zhang K., Liu Q., Wu Y., Yang M. H.*, 2016. Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing* 25, 1779–1792.
- Zhang T., Xu C., Yang M. H.*, 2017. Multi-task correlation particle filter for robust object tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4335–4343.
- Zhang Z., Peng H.*, 2019. Deeper and wider Siamese networks for real-time visual tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4591–4600.
- Zheng W. L., Shen S. C., Lu B. L.*, 2017. Online depth image-based object tracking with sparse representation and object detection. *Neural Processing Letters* 45, 745–758.

ОБ АВТОРАХ ГЛАВЫ

Доктор **Эфстратиос Гаввес** – адъюнкт-профессор Амстердамского университета в Нидерландах, научный руководитель лаборатории глубокого зрения QUVA, научный руководитель лаборатории POP-AART по использованию ИИ для адаптивной лучевой терапии и стипендиат ELLIS. Является получателем гранта ERC Career Starting Grant 2020 и гранта NWO VIDI 2020 для исследования машинного обучения темпоральности пространственно-временных последовательностей. Также является соучредителем Ellogon.AI, дочерней компании университета, в сотрудничестве с Голландским институтом рака (NKI) с целью использования ИИ в патологической медицине и геномике. Эфстратиос – автор нескольких статей для ведущих конференций и журналов по компьютерному зрению и машинному обучению. Ему принадлежат несколько патентов в области компьютерного зрения. Его исследования сосредоточены на машинном обучении и его динамике, эффективном компьютерном зрении и машинном обучении для онкологии.

Дипак Гупта в настоящее время работает научным сотрудником в компании Transmute AI Research в Нидерландах, где занимается фундаментальными исследованиями в области компьютерного зрения и глубокого обучения. Ранее он в течение двух лет работал постдокторантом в лаборатории QUVA и Институте информатики Амстердамского университета, где в основном занимался улучшением методов визуального отслеживания объектов. Дипак защитил докторскую диссертацию в области вычислительной техники в 2017 г., а степень бакалавра и магистра в области геофизики – в 2013 г. Он более года проработал в Royal Dutch Shell научным сотрудником (2017–2019 гг.), решая проблемы геофизики с использованием ИИ. Его особенно интересует разработка эффективных алгоритмов отслеживания объектов и сегментации видео. Кроме того, он также участвует в исследовательских проектах на стыке физики, математики и машинного обучения, ориентированных на применение в медицинской визуализации и геофизике.

Глава 11

Обучение пониманию сцены на основании действий

Авторы главы:

Корнелия Фермюллер, Мэрилендский университет,
Институт передовых компьютерных исследований,
Центр компьютерных наук и технологий Ирибе,
Колледж-Парк, Мэриленд, США;
Майкл Мейнорд, Мэрилендский университет,
факультет информатики, Центр компьютерных наук
и технологий Ирибе, Колледж-Парк, Мэриленд, США

Краткое содержание главы:

- основанный на действиях фреймворк интерпретации сцены и деятельности;
- исследование возможностей и функций объектов и их использования в контексте распознавания действий и обучения роботов;
- исследование распознавания деятельности как взаимодействия познания и восприятия;
- слияние видения и языка через пространства представлений;
- обсуждение перспектив компьютерного понимания действий и деятельности через призму ориентированной на действие структуры.

11.1. ВВЕДЕНИЕ

Целью компьютерного зрения является выработка интерпретаций изображений и видео, которые могут быть полезны людям. Действие следует моделировать, потому что оно является основным средством, с помощью которого люди взаимодействуют с окружающей средой, и в значительной степени благодаря этому взаимодействию окружающая среда становится значимой. Из-за того, что действие занимает центральное место в формировании зна-

чимости окружения, люди выстраивают свое окружение вокруг действия. Таким образом, для полного понимания окружающей среды с точки зрения человека требуется понимание ее отношения к реальным и возможным действиям. Большинство современных методов компьютерного зрения не основаны на действии – в противовес им в этой главе мы представляем методы и фреймворки, которые при моделировании наблюдаемой сцены используют основанную на действии (action based) или функциональную интерпретацию.

Сосредоточение восприятия на действии согласуется с теориями *вплощенного познания* (embodied cognition) (Varela et al., 1993; Barsalou, 2008), которые утверждают, что многие аспекты познания берут свое начало в двигательном поведении и действии. В вычислительном подходе мы можем использовать представление на основе действий в нескольких временных масштабах для иерархического подхода к пониманию сцены. На ранних иерархических уровнях находятся статические компоненты, объекты, люди и простые движения конечностей. Затем они объединяются во все более сложные понятия, включающие взаимодействие между компонентами сцены. *Действия* во времени объединяются структурированным образом в цепочки, образующие *деятельность*.

Использование представлений, основанных на действиях, в методах вычислительного восприятия является сложной задачей. Классический подход к компьютерному зрению заключается в распознавании составляющих сцены исключительно по их внешнему виду. Однако аспекты сцены, связанные с действием, часто носят семантический и реляционный характер и не связаны напрямую с внешним видом физического объекта. Чтобы лучше моделировать взаимодействия, требуются более сложные архитектуры, которые не только моделируют внешний вид, но и используют более осмысленное понимание промежуточной семантической и реляционной структуры действия на входе.

Классические модели со сквозным обучением все хуже работают по мере роста пространства входных состояний, как это происходит в случае видеороликов и действий при увеличении их продолжительности. Это связано с увеличением изменчивости внешнего вида, что создает проблемы как с данными, так и с моделированием. Для масштабирования задачи необходим более когнитивный подход, который моделирует не внешний вид, а структуру действий, составляющих деятельность.

Основным преимуществом подхода к интерпретации сцены на основании действий является *обобщение*, т. е. способность распознавать специфические характеристики сцены помимо тех, которые визуально наблюдаются в обучающей выборке. Например, если мы сможем понять, что делает объект пригодным для резки, это позволит нам распознать новые виды режущих инструментов, такие как аляскинский улу, хотя этот объект отсутствует в нашем обучающем наборе. Точно так же, если мы можем интерпретировать наблюдаемую человеческую деятельность, понимая взаимодействие составляющих действий и выделяя основную цель, мы можем построить более робастную модель. Отдельные составляющие сцены может быть трудно распознать из-за окклюзии, размера, неблагоприятных углов обзора или изменчивости внешнего вида и движений, но рассуждения о когнитивной правдоподобности деятельности помогают исправить ошибки классифика-

ции. Кроме того, моделирование действий дает возможность предсказывать более отдаленное будущее.

В этой главе представлены подходы и концепции, основанные на глубоком обучении и компьютерном зрении, ориентированных на действия. Теперь мы вкратце изложим содержание оставшейся части этой главы.

Раздел 11.2 посвящен *аффордансам*¹ – различным описаниям объектов, основанным на действиях. Аффордансы вызвали большой интерес в области зрения роботов. Но и классическое невоплощенное компьютерное зрение может выиграть от использования аффордансов. Они показывают, как можно использовать различные объекты в сцене, и являются важным компонентом понимания действия. Аффордансы несут информацию о возможных совпадениях наблюдаемых объектов, людей и других составляющих сцены. Раздел 11.2 охватывает наиболее известные работы по этой теме, в том числе ранние исследования, объясняющие аффордансы с помощью геометрических измерений, изучающие карты аффордансов с использованием алгоритмов обнаружения объектов и семантической сегментации на картах глубины и геометрических характеристик, а также исследования, в которых аффордансы сочетаются с другими аспектами для распознавания действия.

Раздел 11.3 посвящен в первую очередь нашей собственной работе по пониманию манипулятивной деятельности. Мы утверждаем, что интерпретация деятельности должна осуществляться как непрерывное взаимодействие между процессами рассуждения и восприятия. Деятельность моделируется иерархически. На нижнем уровне находятся модули объектов, действий, пространственных отношений и т. д., которые на более высоком уровне объединяются посредством грамматической формулы. В этом разделе будут описаны грамматика и избранные модули, формирующие понимание, основанное на действиях.

Раздел 11.4 посвящен методам, которые могут обеспечить более тесную интеграцию внешнего вида, семантических и реляционных ограничений. Мы рассматриваем интеграцию в контексте задачи обучения без ознакомления. Мы начнем с рассказа о простых методах, использующих сконструированные атрибуты, и перейдем к более сложным подходам, включающим слияние языка и видения через общие пространства представления, сбор семантической и реляционной информации.

Далее следует обсуждение того, как эти концепции могут быть применены к пониманию действия и деятельности в разделе 11.5, и выводы в разделе 11.6.

11.2. АФФОРДАНСЫ ОБЪЕКТОВ

Психолог Джеймс Гибсон ввел термин *аффорданс* (Gibson, 1977), обозначающий возможности действия, которые объект предоставляет в рамках фи-

¹ Аффорданс (affordance) – визуально определяемое свойство предмета или объекта окружающей среды, которое позволяет использовать его определенным образом. Например, дверной проем позволяет войти в комнату, а молоток позволяет забить гвоздь. – *Прим. перев.*

зических возможностей человека или животного. Например, нож позволяет человеку «резать», «колоть», «тыкать», «метать» и т. д. В последнее время понятие аффорданса вызывает большой интерес в литературе по когнитивной науке и нейронауке, поскольку данные визуализации мозга показали, что инструменты наблюдения активируют двигательные области мозга (обзор в Martin, 2007). Эту концепцию исследовали в различных областях, включая психологию развития, промышленный дизайн, науку о спорте и взаимодействие человека с компьютером. Было предложено множество интерпретаций понятия аффорданса и мнений о его значении. Большинство исследователей различают «аффорданс» и «функцию», причем первое означает свойства объектов, а второе относится к роли, которую объект играет в достижении какой-либо цели. Например, у ручки чашки есть аффорданс «взять», а ее внутренняя часть реализует функцию «вмещение налитого», в то время как электрическая вилка реализует функцию «питания кухонных приборов» или «зарядки телефона», а водопроводный кран поддерживает функцию «наливания» воды. Однако строгого формального определения не существует, и граница между этими понятиями остается размытой¹.

В этом разделе мы сначала объясним использование аффордансов в компьютерном зрении (раздел 11.2.1). Затем предложим вашему вниманию обзор работ, представленных в различных публикациях: раздел 11.2.2 относится к более ранним подходам, которые использовали геометрические характеристики, вычисленные из трехмерных данных, для классификации аффордансов стульев или предметов повседневного обихода. В разделе 11.2.3 описаны работы по изучению аффордансов объектов и их частей с использованием алгоритмов распознавания компьютерного зрения, применяемых к данным глубины или картам геометрических признаков. В разделе 11.2.4 представлены методы, использующие аффордансы вместе с другими детекторами для распознавания сцен и действий, а также подходы, изучающие аффордансы для воплощенных агентов. Раздел 11.2.5 завершает обзор описанием перспектив дальнейших исследований.

11.2.1. Зачем аффордансы нужны компьютерному зрению?

Взгляд на объекты и сцену с точки зрения аффордансов дает информацию для визуальной интерпретации сцены, которая дополняет классические подсказки и помогает повысить надежность и обобщаемость усвоенных представлений. Эта информация касается «действуемости», которую сцена представляет в нескольких пространственных и временных масштабах, относящихся к объектам, группам объектов и всей пространственно-временной сцене.

¹ Например, можно сказать, что возможность воткнуть вилку в розетку – это аффорданс вилки (и розетки тоже), а подача электричества на бытовой прибор – это ее функция (назначение). В самом деле, мы ведь не создавали вилку только ради возможности воткнуть ее в розетку. Но не всегда разница между аффордансом (возможностью) и функцией (назначением) столь очевидна. – *Прим. перев.*

Следовательно, аффордансы предоставляют информацию и ограничения для понимания сцены как в настоящем, так и в проекции на будущее, тем самым способствуя распознаванию в дополнение к предсказанию, как подробно описано далее.

Модели аффордансов, обученные на наборах объектов, можно перенести на новые категории объектов. Иными словами, если модули распознавания научились узнавать аффорданс, они могут обнаруживать его в объектах, которых раньше не видели, например даже в камне, обладающем нужными свойствами. Это связано с тем, что использование объекта зависит от его физических свойств, таких как его форма, размер, материал и вес (Hermans et al., 2011), и мы можем разработать процессы, которые извлекают эти физические свойства из изображений, карт глубины и других модальностей, не зависящих от ранее встречавшихся категорий объектов. Напротив, классификация объектов на изображениях традиционным способом глубокого обучения не дает понимания того, как визуальные характеристики, такие как аффордансы, связаны с объектом.

Аффордансы предоставляют ценную информацию для понимания визуальных объектов, например для понимания «действительной функциональности» объектов (Hassanin et al., 2018) – скажем, в перевернутую чашку нельзя налить чай, а на сломанном стуле нельзя сидеть (Grabner et al., 2011). Другим примером является поиск подкатегорий классических категорий визуальных объектов, например сортировка стульев для различных целей (Stark, Bowyer, 1991).

Поскольку аффордансы отражают возможные действия, которые можно выполнить с объектом, они несут ценную информацию для прогнозирования будущих действий (Koppula, Saxena, 2015; Qi et al., 2018), так как действия связаны друг с другом во времени. Например, хлебный нож в целом представляет собой аффордансы «взять в руку» и «резать», позволяющие выполнить действие «нарезка хлеба», а оно, в свою очередь, является частью деятельности «подготовка корзинки с хлебом», состоящей из нескольких действий, распределенных во времени в строго определенном порядке. Знание о возможности выполнить действие «нарезка хлеба» позволяет сделать предположение о возможных последующих действиях, например о том, что корзинку с хлебом поставят на стол. Подводя итог, можно сказать, что аффордансы и функциональные возможности на уровне объекта также содержат информацию о возможных взаимодействиях объектов, пространственно-временных отношениях и действиях в более длительных интервалах времени. Моделирование этих отношений для извлечения явных или неявных связей в различных временных масштабах и на разных семантических уровнях абстракции имеет большое значение для задачи понимания деятельности.

Концепция аффордансов занимает центральное место в машинном зрении роботов и в исследованиях парадигмы *активного видения* (Bajcsy, 1988). Автор этой парадигмы выступает за то, чтобы видение систем не считалось пассивным процессом. Биологические системы «двигают глазами, чтобы выбрать то, что они видят» в процессе активного зрения. Точно так же искусственные

воплощенные¹ системы (embodied systems) должны иметь возможность изменять точку зрения своих камер, чтобы выбирать, какую информацию собирать из окружающей среды, поскольку разные точки зрения представляют разную информацию. Идя дальше, парадигма также предполагает, что воплощенные системы должны избегать использования сложных процессов общего зрения для всех целей и обрабатывать только информацию, необходимую для решения поставленной задачи (Fermüller, Aloimonos, 1995). Поэтому, когда робот или искусственная система взаимодействует с объектами, часто бывает эффективнее вычислить, для чего может быть использован объект, т. е. вычислить его аффорданс и как его можно использовать, а не классифицировать объект в соответствии с нашими языковыми представлениями. Таким образом, хотя преимущества аффордансов, обсуждаемые в этом разделе, применимы к классической пассивной формулировке компьютерного зрения, в которой агент не взаимодействует с окружающей средой, большая часть исследований аффордансов посвящена зрению роботов.

11.2.2. Первые исследования на тему аффордансов

Аффордансы связаны с действиями. Как следствие они также основаны на физических величинах, связанных с действием. Например, объект, на котором можно сидеть, или объект, в который можно налить воду, имеют определенные физические характеристики, такие как форма, размер или материал и т. д. Все ранние методы использовали такие явные физически значимые представления в модулях распознавания возможностей.

В ранних исследованиях использовали форму и геометрию. Старк и Бойер (Stark, Bowyer, 1991) предложили первый основанный на аффордансах подход к распознаванию объектов с использованием 3D-моделей САПР в качестве входных данных. Граф знаний, похожий на дерево решений, применялся для классификации стульев и подкатегорий стульев (например, обычный стул, качалка, стульчик для кормления, кресло для отдыха), а листьями графа были процедуры классификации геометрических признаков. Эти признаки включали относительную взаимную ориентацию поверхностей, размер объекта, стабильность и близость поверхностей.

Грабнер и др. (Grabner et al. 2011) выделили поверхности, соответствующие аффордансу «сидеть», сравнив геометрию трехмерной модели человеческого скелета в сидячей позе с геометрией объекта. В число признаков входят расстояние и пересечение сетки сплайнов человека с сеткой объекта. Детектор был оценен на моделях Google Warehouse, а также на реальных 3D-данных, собранных с помощью времяпролетной камеры. Для лучшей точности метод был объединен с классификатором на основе изображений. Аналогичным образом Гупта и др. (Gupta et al., 2011) смоделировали аффордансы в 3D-сценах

¹ Термин *воплощенные* (embodied) в данном контексте можно считать синонимом понятий «очеловеченные», «человекообразные» (в когнитивном смысле). – Прим. перев.

в помещении, обнаружив области пространства, которые позволяют человеку использовать его для одной из трех функций: «лежать», «сидеть прямо» и «сидеть откинувшись». Они также использовали ограничения, основанные на занимаемом трехмерном пространстве и контакте со скелетом человека. Однако их метод может принимать на вход изображения, из которых он сначала получает трехмерную геометрию с помощью методов регрессии, основанных на обучении, таких как (Hedau et al., 2009; Lee et al., 2010).

Германс и др. (Hermans et al., 2011) изучили аффордансы повседневных предметов с помощью промежуточных представлений, которые кодируют визуальные и физические характеристики. Визуальные характеристики включали цвет, дискретную форму и текстуру, а физические характеристики включали вес и размер. В модели использовались стандартные классификаторы; метод был продемонстрирован на семи классах аффордансов в области робототехники.

11.2.3. Обнаружение, классификация и сегментация аффордансов

Задача распознавания аффордансов, связанных с объектами и поверхностями сцены, концептуально сходна с задачей распознавания объектов. В ряде недавних методов для локализации и распознавания аффордансов применяли инструменты обнаружения, классификации, сегментации и семантической маркировки объектов. Однако эти методы обычно применялись не к изображениям, а либо к данным RGB-D, либо к картам признаков, рассчитанным на основе данных о глубине. В этом разделе мы рассмотрим несколько таких подходов.

11.2.3.1. Обнаружение аффордансов по геометрическим признакам

В данном разделе мы рассмотрим исследование (Myers et al., 2015) – первый подход, в котором современные инструменты машинного обучения применялись к геометрическим элементам. В этом разделе подробно описан метод обнаружения аффорданса и связанная с ним вычислительная нагрузка.

В центре внимания исследования были инструменты, которые можно встретить на обычном рабочем месте, в частности обнаружение частей инструментов, связанных с различными аффордансами. Был собран набор данных (набор данных о доступности деталей RGB-D) из 105 кухонных, слесарных и садовых инструментов. Объекты помещали на поворотный столик и снимали камерой Kinect с круговым обзором 360°; всего было подготовлено около 300 кадров для каждого объекта, из которых 10 000 изображений RGB-D были аннотированы на уровне пикселей. На рис. 11.1 показаны примеры объектов для пяти из семи аффордансов, а также аннотация для одного из объектов. Следует отметить, что аффорданс связан с поверхностями, например внутренняя поверхность чашки соответствует аффордансу «вмещать», а внешняя – «обхватывать».



Рис. 11.1 ❖ Примеры объектов из набора данных RGB-D Part Affordance Dataset и пример полнокадрового изображения с размеченным вручную эталоном (внизу справа). Эталонные метки включают ранжирование по множеству аффордансов (Myers et al., 2015)

Из необработанных данных о глубине по фрагментам вычислялись признаки формы, в частности нормаль к поверхности, основная кривизна, индекс формы и дескриптор HoG-Depth (гистограмма градиентов глубины). Исходя из того, что эти признаки являются входными данными, были предложены два подхода к классификации: во-первых, *структурированный случайный лес* (structured random forest, SRF), который создает точечную классификацию; и второй алгоритм S-HMP (superpixel hierarchical matching pursuit, иерархическое сопоставление суперпикселей) (Bo et al., 2013). Последний работает, сначала пересегментируя изображение RGB-D в суперпиксели. Затем, с помощью метода обучения по словарю, признаки формы разреженно кодируются в нескольких масштабах на суперпиксель. Наконец, выполняются тах-пулинг признаков в суперпиксели и классификация с помощью SVM. Примеры результатов показаны на рис. 11.2 как для метода S-HMP, так и для метода SRF, где уровень серого отражает вероятность присвоения аффорданса.

В упомянутом подходе есть два вычислительных аспекта, которые заслуживают особого внимания. Во-первых, что иногда упускают из виду, назначение аффордансов поверхностям объектов в целом не может быть уникальным. Одна и та же часть объекта может использоваться для разных целей. Таким образом, присвоение аффордансов является задачей *многоклассовой разметки*. Майерс и др. (Myers et al., 2015) решили эту проблему за счет того, что несколько экспертов-аннотаторов ранжировали, насколько близки другие аффордансы по отношению к рассматриваемым базовым аффордансам, на основании чего была получена порядковая шкала присвоения аффордансов при тестировании.

Во-вторых, основным преимуществом подхода является его хорошее обобщение на новые объекты и поверхности. На рис. 11.2 (внизу) можно увидеть, что дно чашки ассоциируется с аффордансом «ударять», а край шпателя — с аффордансом «резать». Так происходит, потому что форма объектов указывает на эти свойства. Но одной лишь формы было бы недостаточно для классификации аффордансов в реальной прикладной системе. Необходимо

добавить дополнительные свойства, самое очевидное из которых – материал. Благодаря этому система сможет понять, что бумажный стаканчик нельзя использовать для забивания гвоздей, а предмет с мягким краем нельзя использовать для резки.

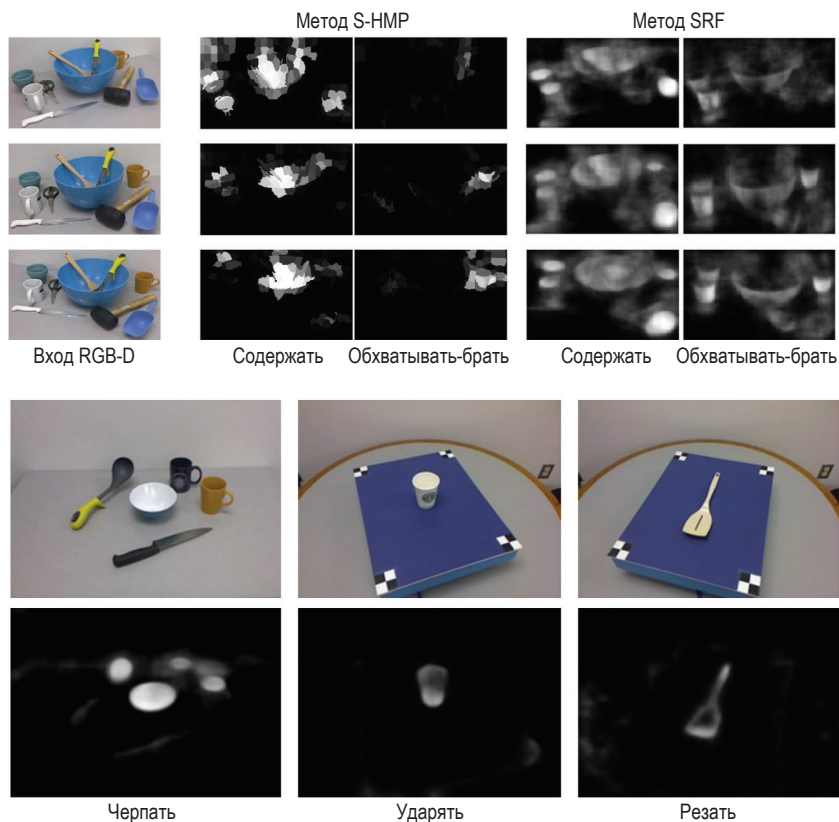


Рис. 11.2 ❖ Вверху: результаты обнаружения аффордансов в трех входных кадрах RGB-D с использованием метода S-HMP и SRF в загроможденной последовательности для целевых аффордансов «вмещать» и «обхватывать-брать». Более яркие пиксели означают более высокую вероятность целевого аффорданса (Myers et al., 2015). Внизу: демонстрация обобщения на новые объекты для метода SRF: дну чашки присвоена высокая вероятность аффорданса «ударять», а краю шпателя – «резать»

11.2.3.2. Семантическая сегментация и классификация по изображениям

Многие из работ, последовавших за (Myers et al., 2015), использовали нейросетевые подходы, применяя в качестве входных данных карты геометрических объектов. Таким образом, обнаружение аффордансов на уровне пикселей превратилось в задачу семантической маркировки. Однако, в отличие























от исходного метода, эти подходы часто использовали в качестве входных данных 2D-изображения. На этапе предварительной обработки карты глубины или карты объектов регрессировали с помощью нейронных сетей. Кроме того, некоторые исследователи рассматривали естественные изображения с несколькими объектами и использовали алгоритмы обнаружения объектов для локализации объектов, прежде чем назначать аффордансы.

Например, Нгуен и др. (Nguyen et al., 2017) создали набор данных с десятью категориями объектов и девятью категориями аффордансов из области домашнего хозяйства и мастерской. Он состоит как из сканов RGB-D, так и из естественных изображений (подмножество ImageNet (Russakovsky et al., 2015)) – для последних карты глубины были созданы с использованием CNN-подхода Liu et al. (2015). Изображения были аннотированы ограничивающими рамками и аффордансами на уровне пикселей. В методе из упомянутой статьи сначала применялся детектор объектов, затем в каждом регионе вычислялись аффордансы с использованием модифицированной сети VGG-16, обученной семантической маркировке, и, наконец, значения аффордансов подвергались постобработке с помощью CRF.

Шриканта и Галл (Srikantha, Gall, 2016) использовали набор данных Коппулы и Саксены (Koppula, Saxena, 2014), который содержит богатую контекстную информацию с точки зрения взаимодействия человека с объектом, и разметили его с помощью аннотаций аффордансов на уровне пикселей. В работе изучались различные уровни обучения семантической сегментации с использованием глубокой сверточной нейронной сети в рамках схемы максимизации ожиданий, чтобы использовать в качестве контекста слабо размеченные данные, такие как аннотации уровня изображения или аннотации ключевых точек, а также позу человека.

Рой и Тодорович (Roy and Todorovic, 2016) работали со сценами в помещении из набора данных Нью-Йоркского университета (Silberman et al., 2012). Их метод сначала выводит карту глубины, нормали поверхности и семантическую сегментацию грубого уровня с использованием многомасштабной CNN в качестве сигналов среднего уровня, которые затем совместно подаются в качестве входных данных в другую многомасштабную CNN для прогнозирования карт аффордансов.

Йе и др. (Ye et al., 2017) разработали метод локализации и распознавания функциональных зон в сценах внутри помещений. Для категоризации областей изображения была определена онтология, как показано на рис. 11.3 (слева), в соответствии с их аффордансом или функциональностью. Категории включают в себя: «открыть сферическим захватом» (например, дверная ручка), «открыть полным захватом или перетащить, чтобы открыть» (например, дверца духовки), «включить электричество» (например, выключатель света) и т. д., как показано в предпоследнем столбце рисунка. Набор данных содержит 500 изображений кухонь из набора данных SUN (Xiao et al., 2010), которые были тщательно размечены. Метод сначала запускает детектор на основе CNN, обученный обнаруживать область, а затем классификатор на основе архитектуры VGG. На рис. 11.3 (справа) показаны примеры результатов.

	Функцион. область	Основная функция	Конечная категория	Символ	Цветовой код
Онтология функциональной области	Небольшая часть мебели/прибора/стены	Открыть	Сферическая хватка для открывания		
			Обхват для открывания		
		Включить/выключить	Вкл./выкл. электричество		
	Предметы (сосуды и инструменты)		Вкл./выкл. воду		
			Вкл./выкл. огонь		
		Переместить	Две руки поднимаются и двигаются		
			Цилиндрический захват для перемещения		
			Подцепить для перемещения		
			Зажать, чтобы двигать		
	Мебель	Манипулировать	Работа с удлиненными инструментами		
		Использование мебели	Сесть, поставить и т. д.		

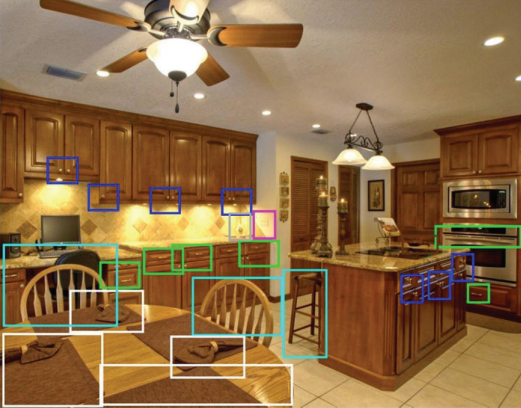


Рис. 11.3 ❖ Слева: функциональная онтология; справа: пример результатов обнаружения (Ye et al., 2017)

11.2.4. Аффорданс в контексте распознавания действий и обучения роботов

В этом разделе мы выделяем несколько подходов, в которых применяют аффордансы в сочетании с другими величинами для понимания сцены и действия. Затем обсудим подходы, направленные на изучение аффордансов роботами.

11.2.4.1. Распознавание действий

Аффордансы кодируют признаки возможных взаимодействий человека с окружающей средой. Следовательно, они естественным образом обеспечивают связку между различными свойствами сцены в пространстве-времени, например между различными объектами или между объектами и действиями. Ряд исследований опирался на эту идею и использовал отношения между аффордансами в качестве контекста для распознавания и прогнозирования активности и действия. В этих методах использовались различные модели для кодирования взаимосвязей между разными вовлеченными объектами, включая CRF, MRF, And-Or-Graphs и вероятностные автоматы состояний.

Кьеллстрём и др. (Kjellström et al., 2011) исследовали проблему изучения взаимодействий действие–объект на демонстрационном примере, в котором они задействовали аффордансы. Действия рук были классифицированы в контексте объектов, которыми манипулируют, с использованием модели CRF, получающей в качестве входных данных объект и признаки рук. Объекты были смоделированы с использованием созданных вручную признаков, а действия – с помощью общей скорости руки, ориентации и углов сустава, которые были рассчитаны на основе результатов трехмерной реконструкции руки и метода отслеживания.

Коппула и др. (Koppula et al., 2013) рассмотрели проблему изучения последовательностей субактивных (subactivity, группы действий), выполняемых

людьми, и их взаимодействия с объектами. Они совместно смоделировали человеческую деятельность и аффордансы объектов в марковском случайном поле, где узлы представляют объекты и субактивности, а ребра – отношения между аффордансами объектов, их отношения с субактивностями и их эволюцию во времени. Отношения аффорданс–субактивность рассчитывались на основе относительных геометрических характеристик между объектом и суставами скелета человека, а аффордансные отношения между объектами – на основе пространственных отношений. Описанный подход был продемонстрирован на примере робота PR2 при выполнении вспомогательных задач. Коппула и Саксена (Koppula, Saxena, 2015) добавили в марковскую модель также возможные будущие состояния, чтобы предсказать следующее действие.

Ци и др. (Qi et al., 2017) использовали пространственно-временной граф «И-ИЛИ» (spatial-temporal AND-OR Graph, ST-AOG) для представления структуры деятельности и прогнозирования будущих действий во входном видеосигнале RGB-D. Их модель иерархична: субактивности моделируются человеческими действиями, объектами и их аффордансами в пространственных графах, а стохастическая грамматика, определенная для субактивностей, кодирует деятельность. Дутта и Зелинска (Dutta, Zielinska, 2017) также рассмотрели проблему прогнозирования следующего действия на основе аффордансов объектов и человеческого взаимодействия. Они использовали пространственно-временные вероятностные автоматы состояния для моделирования взаимодействий. В дополнение к классу действия они также вычислили возможную траекторию действия. В зависимости от того, где объект находится относительно человека, он имеет разные аффордансы, и его ориентация и расстояние до возможных траекторий действий кодируются в виде карт интенсивности в зависимости от аффорданса.

11.2.4.2. Изучение аффордансов в зрении роботов

Понятие аффорданса было ключевой концепцией в области нейроробототехники, целью которой является изучение когнитивных функций робототехнической системы, тело которой помещено в окружающую среду. Речь идет о том, что роботы приобретают все более сложные навыки, используя восприятие и взаимодействие с окружающей средой. Благодаря взаимодействию роботы изучают аффордансы и строят на их основе иерархическое понимание действий, деятельности и окружающей среды. Это исследование в области *саморазвивающейся робототехники* стало возможным благодаря разработке роботизированных платформ, наиболее известной из которых является человекоподобный робот iCub (Metta et al., 2008).

В работе (Fitzpatrick et al., 2003) авторы обсудили три основных этапа саморазвития робота: (1) изучение образа тела, (2) изучение взаимодействия с внешними объектами и (3) изучение интерпретации взаимодействия объект–объект. Аффордансы занимают центральное место на последних двух этапах. Робот-гуманоид посредством толкающих и тянущих действий в разных направлениях научился взаимодействовать с окружающей средой, а наблюдая за движениями затронутых объектов, изучил аффордансы. Например, он

узнал, что сферический объект может катиться, а прямоугольный – скользить. Наконец, робот также научился повторять наблюдаемое действие.

Точно так же авторы Montesano et al. (2008) определили три основных этапа в архитектуре развивающегося робота-гуманоида: сенсорно-моторную координацию, взаимодействие с миром и подражание. Аффордансы играют центральную роль во взаимодействии с миром. При таком подходе система начинала с базовых зрительных и моторных навыков, из которых с помощью алгоритмов кластеризации приобретались более сложные зрительные и моторные навыки. Затем во время взаимодействия наблюдались эффекты с помощью восприятия, такие как изменения положения объекта, скорости и тактильного восприятия. Байесовская сеть использовалась для изучения аффордансов, которые в данном случае были закодированы как вероятностные отношения между действиями и восприятиями (характеристики объекта и эффекты). Было продемонстрировано, что система имитирует действия людей, выполняя движения с аналогичным эффектом.

Угур и др. (Ugur et al., 2011) также продемонстрировали возможность обучения робота объектам посредством взаимодействия и самонаблюдения. На первом этапе робот обнаружил общие черты в своих действиях и эффектах, обнаружив категории эффектов. Опираясь на это, на втором этапе были получены предикторы аффордансов для различного поведения путем изучения сопоставления характеристик объекта с категориями эффектов. В следующей работе Угур и Пиатер (Ugur, Piater, 2016) пошли еще дальше и изучили механизмы, обеспечивающие иерархическое структурирование задач изучения аффордансов. Руководствуясь внутренними соображениями, робот начал с простых задач и, опираясь на свои знания о взаимодействиях, постепенно осваивал более сложные задачи, выбирая для исследования объект и действие, наиболее отличающиеся от ранее изученных. В экспериментах робот мог вычислять визуальные характеристики размера объекта, форму участка поверхности и нормали к поверхности, а его действия заключались в том, чтобы тыкать в объекты с трех разных направлений и складывать их друг на друга. На более ранних этапах он исследовал действие тыкания, чтобы наблюдать его влияние на отдельные объекты. Опираясь на полученные знания, на следующем этапе он исследовал размещение одного объекта поверх другого и возникающие в результате эффекты.

11.2.5. Промежуточный итог – изучение аффордансов

В этом разделе обсуждались подходы к изучению аффордансов, многие из которых относятся к области зрения роботов и были реализованы с небольшим количеством образцов и ограниченным объемом данных. До сих пор в исследованиях, посвященных аффордансам, относительно мало использовали подходы глубокого обучения. Основная причина – отсутствие больших аннотированных наборов данных в этой области, необходимых для глубокого обучения.

Однако мы ожидаем, что по мере того, как исследования будут продвигаться от глубокого обучения с учителем к методам обучения без учителя

и самообучения, мы увидим подходы, основанные на концепции аффордансов и наблюдаемых взаимодействий между людьми и объектами. Этому будут способствовать новые специализированные наборы данных, такие как набор данных EPIC Kitchens, в котором представлены различные действия по манипулированию объектами в естественных сценах (Damen et al., 2018).

Аффордансы и функциональные возможности на уровне объекта также включают в себе информацию о возможных взаимодействиях объектов, пространственно-временных отношениях и возможных действиях в более длительной перспективе. В разделе 11.2.4 мы обсудили методы, использующие аффордансы для моделирования действий. Однако в дальнейшем имеет смысл построить модель отношений, чтобы получить явные или неявные отношения в более длительных масштабах времени для решения задачи понимания деятельности, о которой пойдет речь в разделе 11.3.

Наконец, мы можем использовать аффордансы при создании отображений от восприятия к действию для обучения роботов. Люди могут научиться манипулятивным действиям, используя только свое восприятие. Когда мы видим, что кто-то выполняет действия с незнакомым нам инструментом, мы можем понять возможности этого инструмента и выполнить то же самое действие. Точно так же мы могли бы подойти к обучению моторики роботов, используя восприятие и действие в тесной петле, обосновывая их аффордансами, чего еще не делалось раньше. Робот будет изучать задачу, наблюдая за действием и аффордансами и выдавая команды (на основе своего существующего набора навыков, ограниченного аффордансами), чтобы генерировать действие, близкое к наблюдаемому, а затем постепенно адаптироваться для улучшения качества. Предложенная исследовательская задача фактически представляет собой разработку методов самообучения и обучения с подкреплением, на основании представлений аффордансов.

11.3. ФУНКЦИОНАЛЬНЫЙ АНАЛИЗ МАНИПУЛЯЦИЙ

В этом разделе описывается работа – в основном нашей исследовательской группы – по интерпретации манипулятивной деятельности. Опираясь на парадигму воплощенного познания (Варела и др., 1993), в этой работе мы рассматриваем понимание человеческой деятельности как процесс, включающий восприятие, познание и двигательную систему. Основными компонентами являются формализованные способы объединения различных модальностей и модули компьютерного зрения для получения семантически значимых дескрипторов действия.

11.3.1. Активное взаимодействие между познанием и восприятием

Понимание человеческих действий и деятельности является наиболее сложной задачей, изучаемой в настоящее время в области компьютерного зрения.

Эта задача относится не только к видению. Люди могут понять, что делают другие, потому что у них есть модели действий и деятельности. Они понимают цели действий, и это позволяет им интерпретировать свои наблюдения, несмотря на обширный спектр вариантов действий и условий, в которых они наблюдаются. Знание (в той или иной форме) включается в процесс толкования на раннем этапе.

Человеческое поведение является активным и исследовательским. Мы постоянно перемещаем взгляд в разные места сцены. Мы распознаем объекты и действия, и это, в свою очередь, заставляет нас фиксировать внимание на новых местах. В этом процессе восприятие непрерывно взаимодействует с познанием на разных уровнях абстракции, чтобы направлять внимание, делать прогнозы, ограничивать пространство поиска для распознавания и рассуждать о том, что воспринимается. Мы называем это взаимодействие между непосредственным восприятием и процессами более высокого уровня *когнитивным диалогом* (Aloimonos, Fermüller, 2015), поскольку оно представляет собой итерацию вопросов и ответов, при этом когнитивные или лингвистические процессы задают вопросы о том, что и где находится на сцене, а визуальные процессы выполняют локализацию, обнаружение, распознавание и реконструкцию. Одним из простых способов выбора следующего вопроса может быть использование теоретико-информационных критериев (Yu et al., 2011).

Логические построения могут быть реализованы с помощью стратегии, основанной на знаниях (Aditya et al., 2018), или с использованием языка. В этом плане компьютерное зрение вызывает особый интерес, поскольку оно позволяет ввести в процесс интерпретации дополнительные знания об отношениях изображений на более высоком уровне. В то время как в одних исследованиях извлекают эту дополнительную информацию из заголовков или сопроводительного текста, в других (как будет сказано в разделе 11.4) используют расширенную обработку естественного языка для получения дополнительной информации высокого уровня. В текущих исследованиях в качестве языкового представления чаще всего используется пространство word2vec (Mikolov et al., 2013) (раздел 11.4.3), которое кодирует сходство лингвистических понятий. В качестве альтернативы можно использовать старые, созданные вручную ресурсы, кодирующие лексическую семантику, например базу данных Word-Net (Miller et al., 1990), которая связывает слова через *синонимию* (слова, имеющие одинаковое значение, например «бить» и «ударять») и *гипернимию* (отношения «представляет собой», как, например, между «автомобиль» и «транспортное средство»). В этом отношении особенно интересна сеть Verbnets (Schuler, 2005), организующая классы глаголов для понимания действия.

11.3.2. Грамматика действий

Для кодирования отношений между различными семантическими понятиями, то есть между действующими лицами, объектами, глаголами, пространственно-временными отношениями и атрибутами, традиционно при-

меняются различные механизмы. В разделе 11.2 обсуждалось использование графов «И-ИЛИ» и марковских моделей. К альтернативным механизмам относятся логические сети Маркова (Tran, Davis, 2008) и средства планирования (Guha et al., 2013). В этом разделе мы описываем применение *грамматик* (grammar), которые могут фиксировать состав наблюдаемых действий в виде последовательностей, составляющих сцены и их рекурсивную структуру.

Основная причина использования грамматик связана с идеей о том, что наблюдаемые в видео действия имеют синтаксическую структуру. Зная цель действий, видео можно разбить на осмысленные сегменты, а эти сегменты можно организовать в виде простой грамматики. Иными словами, интерпретация действия, происходящего в видео, подобна пониманию предложения, которое мы читаем или слышим. Чтобы разбить видео на примитивные действия, составляющие сложные задачи, сегменты видео сопоставляются с определенными символами, включающими объекты, инструменты, движение и пространственные отношения. Важно отметить, что грамматика действия при сегментации видео исходит из понятия *контакта*, то есть момента времени, когда рука касается объекта или отпускает его, или когда объекты сливаются или разделяются. В эти моменты начинается новое поддействие. При применении грамматики для анализа видео создается дерево синтаксического анализа, которое мы называем *деревом деятельности* (activity tree). Эта концепция наглядно представлена на рис. 11.4. Из видеозаписи человека, выпол-

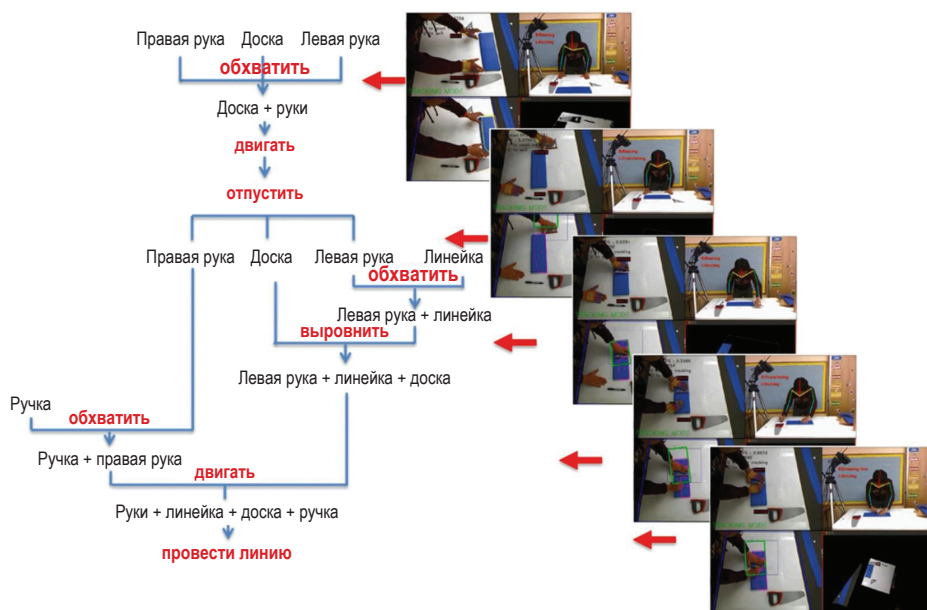


Рис. 11.4 ❖ Иллюстрация описания деятельности. Камера следила за тем, как человек отпиливает доску. Четыре параллельных процесса вычисляют основные компоненты: обнаружение руки и классификация типа захвата (слева вверху), грубое определение движения посредством подгонки скелета (справа вверху), сегментация объекта (слева внизу), трехмерное описание сцены (справа внизу)

няющего действие «отпиливание доски», создается граф, в котором узлы рук, предметов и инструментов сливаются в общий узел, когда они соприкасаются, или узлы расходятся, когда предметы и руки расходятся. Представленные на рисунке различные процессы (показаны в четырех квадрантах видеокадров) извлекают тело человека, руки, объекты и геометрические отношения.

Далее мы рассмотрим несколько грамматических методов в разделе 11.3.2.1, а затем обсудим в разделе 11.3.2.2, являются ли такие грамматические представления достаточно выразительными, чтобы уловить действие и структуру деятельности, и достаточно ли они экономичны в вычислительном отношении, чтобы их можно было предпочесть другим представлениям.

11.3.2.1. Различные реализации грамматики

Описания основаны на *контекстно-независимых грамматиках* (context-free grammar), первоначально представленных в (Pastra, Aloimonos, 2012). Саммерс-Стей и др. (Summers-Stay et al., 2012) реализовали идею анализа комплекса действий из видео RGB-D с использованием только одного символа для всех действий, а (Yang et al., 2014) расширили описание, включив в него «захват» и разграничив контакт «рука–объект» и контакт «объект–объект». Грамматика описывает действия на уровне абстракции, который полезен как для интерпретации видео, так и для выполнения действий роботом. В работе (Yang et al., 2015) это было продемонстрировано на нескольких примерах. Путем автоматического анализа видео с инструкциями по приготовлению пищи из набора данных Youcook (Das et al., 2013) были проанализированы действия, которые затем были выполнены роботом Baxter, обладающим необходимыми двигательными возможностями.

Абстрактные описания действий/глаголов необходимы для достижения обобщения и выполнения того, что в современной терминологии называется обучением за несколько шагов, или обучением без ознакомления (см. раздел 11.4). Базовая грамматика действия сводит описание действий только к последовательности «отношений прикосновения», то есть когда рука касается предмета, соприкасаются два предмета, рука отпускает предмет или когда два предмета или части одного предмета разделяются (Dessalene et al., 2021).

Бёргёттер и др. (Wörgötter et al., 2013) уточнили эту концепцию, чтобы сформулировать онтологию с учетом действий одной рукой. На первом уровне действия классифицируются по шести классам в соответствии с последовательностью отношений, которые могут иметь рука и один или два объекта: переставить, уничтожить, сломать, опрокинуть, спрятать, создать. Исходя из этих классов, они повторяют возможные действия и придумывают около 30 основных манипуляций. Янг и др. (Yang et al., 2013) предложили родственную концепцию. Они разбили действия на *метаклассы* в соответствии с последствиями действия для объекта, то есть тем, что происходит с объектом геометрически или топологически. Они предложили шесть категорий – разделить объект, объединить две части, перенести объект, деформировать объект, объект появляется, объект исчезает со сцены, – а также предоставили алгоритмы, которые сочетают отслеживание с сегментацией для обнаружения топологических изменений, свидетельствующих об основных событиях в видео.

11.3.2.2. Являются ли грамматики выразительными и лаконичными описаниями?

Важный вопрос заключается в том, действительно ли грамматические представления достаточно богаты, чтобы можно было классифицировать многие виды деятельности. Авторы статьи (Wörgötter et al., 2020) провели психофизические и вычислительные эксперименты, чтобы ответить на этот вопрос. Они описывали, как и выше, действия последовательностью контактов, используя пять величин: «рука», «земля» и три предмета. Кроме того, они рассмотрели десять пространственных отношений, то есть «вверху», «внизу», «между» и т. д., чтобы различать в общей сложности 35 различных конфигураций или действий. Подмножество из десяти этих действий (*положить, встряхнуть, перемешать, взять, открыть, нарезать, разрезать, спрятать, положить и толкнуть*) выполнялось в виртуальной среде, но вместо реальных объектов использовались кубы. Эксперименты показали, что исследуемые алгоритмы могут распознавать эти действия так же, как и люди. Более того, предложенное описание оказалось очень мощным в предсказательном плане – испытуемым в среднем требовалось только 56 % продолжительности описания, чтобы распознать действие. Таким образом, можно заключить, что описание, основанное только на контактах и пространственных отношениях, очень эффективно для визуального распознавания действий и деятельности.

11.3.3. Модули для понимания действий

Для распознавания дискретных компонентов сцены необходимы отдельные процессы зрения, которые затем можно объединить в процессы рассуждений более высокого уровня, такие как грамматики из раздела 11.3.2, чтобы реализовать распознавание и предсказание деятельности. Deskriptory, которые мы обсуждаем в этом разделе, отличаются от широко описанных в литературе (обзор успешных концепций, применяемых в текущих методах, представлен в (Sigurdsson et al., 2017)). В частности, в разделе 11.3.3.1 мы обсуждаем представления захватывания, а в разделе 11.3.3.2 – явное представление геометрии.

11.3.3.1. Захватывание: важный признак для понимания действий

Тип захвата предоставляет важную информацию о действиях. В качестве иллюстративного примера рассмотрим две сцены на рис. 11.5 из задачи VOC (Everingham et al., 2010). Стандартные системы компьютерного зрения имеют детекторы объектов и людей, достаточные для распознавания велосипеда и велосипедиста, а также детекторы позы для подтверждения того, что эти два велосипедиста едут на велосипеде. Но люди могут сказать, что велосипедист с левой стороны не участвует в гонке (поскольку его руки находятся в положении «отдых или разгибание»), тогда как велосипедист справа явно

участвует в гонке (поскольку руки крепко держат руль в «сжатом» положении типа «силовой цилиндрический захват»).



Рис. 11.5 ❖ Упираие разогнутыми руками в руль (слева) в сравнении с силовым цилиндрическим захватом руля (справа) (Yang et al., 2015)

Здесь мы рассмотрим две статьи: в первой используется базовая онтология типов захватов в задачах понимания действий, во второй изучаются тонкие изменения в захватах для различения похожих манипуляционных действий, а также разрабатываются подходы к обучению для прогнозирования действий в реальном времени и регрессии связанных усилий пальцев.

Распознавание типа захвата дает важную информацию для более подробного анализа действия (рис. 11.6). Исследователи в нескольких областях, включая робототехнику, медицину и биомеханику, разработали таксономии захвата, которые представляют собой иерархию наиболее распространенных поз рук, используемых для захвата объекта, причем каждая таксономия основана на потребностях решаемых задач в предметной области. В работе (Yang et al., 2015) использовалась базовая классификация основных функциональных захватов (Cutkosky, 1989) в манипуляционных задачах, затем представленная как полезная функция в двух других задачах: для сегментации действий, связанных с мелкой моторикой, и для характеристики намерения действия, т.е. когда задача случайная либо требует навыков или сил.

Когнитивные исследования показали, что намерение *актора*¹ формирует кинематику его движения во время выполнения движения (Ansuini et al., 2015). Например, когда испытуемые брали бутылку, чтобы налить из нее жидкость, средний и безымянный пальцы были более вытянуты, чем когда они брали бутылку с намерением сдвинуть, бросить или передать ее. Вдохновленные этими выводами, Фермюллер и др. (Fermüller et al. (2018) разрабо-

¹ Актор – действующее лицо сцены; субъект, выполняющий интересующее нас действие. – *Прим. перев.*

тали архитектуру рекуррентной нейронной сети, которая отслеживает руки для прогнозирования действий. В частности, они рассматривали наборы действий с одним и тем же предметом, такие как «отжимание», «переворачивание», «умывание», «вытирание» и «царапание» губкой (см. рис. 11.7). Они проанализировали систему, которая в режиме реального времени предсказывала текущие действия, чтобы определить, в какой момент времени классификация стала точной, а также провели психофизический эксперимент, оценивая эффективность человека при выполнении той же задачи. На 10 кадрах после контакта руки с объектом система и люди начали понимать действие (75%-ная точность классификации действий губкой), а на 25 кадрах оценка была очень хорошей (точность 95 %). Архитектура визуальной сети представляла собой RNN, использующую в качестве входных данных отслеживаемые участки изображения вокруг руки, на основе которых были вычислены признаки VGG-16 (Simonyan, Zisserman, 2014).



Рис. 11.6 ❖ Базовая классификация активных захватов с примерами (Cutkosky, 1989). На самом высоком уровне захваты подразделяются на силовые и точные. Силовые захваты используются, когда объект удерживается с силой, и могут быть классифицированы как цилиндрические, сферические и крюкообразные. Точные захваты обеспечивают точное движение и подразделяются на обхватывающие, трехпальцевые и червеобразные

Кроме того, в статье также продемонстрирована связь зрения с усилиями. Были зарегистрированы данные об испытуемых, которые выполняли одно и то же действие обеими руками. На пальцах одной руки усилия регистрировали при помощи датчиков, а другой руки – визуально. Рекуррентная нейронная сеть была обучена регрессии усилия по визуальным данным. Затем было показано, что, используя только входное видео, когда визуальный классификатор сочетается с регрессией усилия, можно достичь улучшенной точности. Нам этот подход представляется многообещающим. Как показано в статье, изучение сопоставления зрительной картины с картой усилий

формирует бимодальное пространство, которое помогает визуальному распознаванию. Кроме того, у этой концепции есть прямое применение в робототехнике. В настоящее время для изучения задач роботы используют тактильные устройства или датчики силы и крутящего момента. Если мы сможем визуальным образом предсказать силы, прилагаемые человеком-демонстратором, это позволит нам гораздо эффективнее обучать роботов.



Рис. 11.7 ❖ Примеры, доказывающие, что начальные движения могут быть надежными индикаторами предполагаемых манипуляций. Раннее предсказание действий значительно снижает задержку взаимодействия в реальном времени, что принципиально важно для проактивной системы

11.3.3.2. Геометрические факторы для робастизации

Использование геометрических факторов важно, поскольку они предоставляют надежную информацию для описания сцены и дополнительную информацию для распознавания. Геометрия объектов сцены вычисляется с использованием процессов реконструкции, которые являются низкоуровневыми (требуются только признаки изображения и знание положения камеры), а обучающие данные или процесс машинного обучения не требуются. С появлением доступных по цене сенсоров RGB-D более десяти лет назад воссоздание геометрии сцены стало намного проще и точнее, и поэтому эти сенсоры стали стандартными датчиками зрения в робототехнике. Их использование облегчает точное и быстрое вычисление расстояний для управления движением робота, а также вычисление геометрии и формы объектов для облегчения интерпретации сцены. В этом разделе обсуждаются три геометрических метода: точное отслеживание трансформаций нежестких объектов и обнаружение топологических изменений, вычисление попарных пространственных отношений объектов во времени и вычисление симметрии объекта и ее использование для лучшей сегментации переднего плана и фона.

Опираясь на эффективную библиотеку облаков точек (Zampogiannis et al., 2018), в своей следующей работе (Zampogiannis et al., 2019) авторы предложили методику точного отслеживания трансформаций нежестких объектов и обнаружения топологических изменений, то есть контактов и разделений частей тела и объектов, необходимых для грамматического описания (раздел 11.3.2). Суть метода заключается в оценке поля деформации, которая учитывает деформации между последовательными кадрами для обнаружения областей деформированной геометрии, которые претерпевают топологические изменения.

В описании деятельности также могут пригодиться дескрипторы пространственных отношений между объектами. В статье (Zampogiannis et al., 2015) авторы ввели представление манипуляционных действий, основанное на эволюции пространственных отношений между объектами в сцене. Метод был реализован путем отслеживания объектов в видео RGB-D и выводов о пространственных отношениях наблюдаемых пар объектов. Результирующий дескриптор представляет собой последовательность предикатов пространственного отношения (например, *внутри, слева, справа, впереди, сзади, внизу, вверху, касание*). Было показано, что выразительности этого дескриптора достаточно для различения четырех различных действий.

Другая концепция заключается в использовании общих знаний о свойствах формы объекта. Например, обнаружение симметрии может помочь в сегментации как в 2D (Teo et al., 2015), так и в 3D (Ecins et al., 2016). Представьте, что вы смотрите на загроможденную сцену. Поскольку большинство объектов, с которыми мы работаем, симметричны либо зеркально, либо вращательно, мы можем «дополнить» невидимую часть объекта, что существенно помогает сегментации и распознаванию.

11.3.4. Проблематика понимания деятельности

Понимание деятельности – очень сложная задача. Законченные сквозные решения плохо масштабируются из-за больших различий во внешнем виде на высоких уровнях абстракции и временного расширения.

В предыдущих разделах мы рассмотрели иерархические подходы и подробно описали один более высокий уровень – грамматики действия. Грамматики действий могут сегментировать действия во время контакта и фиксировать рекурсивную структуру последовательностей действий, аналогичную той, что встречается в языке. Мы описали эксперименты, демонстрирующие выразительную силу грамматик действия. Мы также описали процессы нижнего уровня, которым не уделялось должного внимания в компьютерном зрении, но которые необходимы для реализации методов, основанных на действиях. К ним относятся аффордансы, анализ типа захвата и использование геометрии для облегчения временной и пространственной сегментаций и описания пространственных отношений.

В этом разделе мы подчеркнули необходимость использования осмысленных представлений, основанных на действиях, на разных уровнях иерархии. Точнее говоря, очень важно, чтобы эти представления были робастными

из-за многих проблем, связанных с пониманием деятельности. Частично мы можем добиться робастности за счет использования геометрии, поскольку она не требует запоминания и может быть оценена по низкоуровневым измерениям. Следовательно, прежде чем начинать распознавание, мы должны по возможности вводить в систему компьютерного зрения геометрию. Помимо геометрии, для понимания деятельности имеет значение любое понятие, которое предоставляет универсально верную информацию. Мы можем моделировать физические законы или включить модельные онтологии, чтобы облегчить обобщение, например сгруппировав глаголы в зависимости от того, какое влияние они оказывают на объекты (Yang et al., 2013). Мы также можем добавить процессы, моделирующие причинность; действия причинно ограничивают друг друга, некоторые комбинации невозможны физически. Эти представления несут больше знаний и лучше ограничивают интерпретацию деятельности.

Интеграция зрения и познания или языка сложна. Это происходит из-за *семантического разрыва*, то есть несоответствия между символическими или языковыми представлениями и визуальными представлениями, основанными на сигналах. Мы стремимся добиться устойчивой интеграции между этими представлениями. Поэтому нам нужно избегать слишком ранней настройки пороговых значений или преобразования в чисто символическое представление. Это связано с тем, что если зрение не может предоставить точные данные, то неточности усугубляются дальнейшей абстракцией, что приводит к ошибочным рассуждениям. Следующей исследовательской задачей является выработка способов обучения, которые связывают восприятие с рассуждениями более высокого уровня для более глубокой интеграции. В разделе 11.4 рассматриваются подходы к глубокому обучению, полезные для такой интеграции. В настоящее время подобные методы в основном ограничены распознаванием объектов и, в некоторой степени, распознаванием действий, но они будут полезны и для задачи распознавания деятельности.

11.4. ПОНИМАНИЕ ФУНКЦИОНАЛЬНОЙ СЦЕНЫ ПОСРЕДСТВОМ ГЛУБОКОГО ОБУЧЕНИЯ С ПОМОЩЬЮ ЯЗЫКА И ЗРЕНИЯ

В этом разделе мы рассматриваем слияние видения и языка и связанных с ними представлений – слияние «сигнала» и «символа». Объединение нескольких представлений важно из-за пригодности различных представлений для отражения различных характеристик мира. Мы стремимся к тому, чтобы информация из представлений внешнего вида более низкого уровня и информация из представлений отношений и семантики более высокого уровня дополняли друг друга.

Системы, включающие символические и непрерывные представления, часто имеют жесткие границы, ниже которых система непрерывна, а выше –

символична. Где именно устанавливается эта граница, зависит от ситуации, но обычно она не опускается ниже уровня абстракции, отраженного в человеческом языке, поскольку язык является основным источником символически представленного знания о мире.

Символическое представление более важно для понимания действий и деятельности, чем для других задач компьютерного зрения, таких как обнаружение объектов, поскольку природа этой задачи более абстрактна и менее основана на внешнем виде. Действие имеет временную структуру в нескольких масштабах и базируется на удовлетворении условий – определяющими характеристиками действия являются семантика и отношения.

Многие задачи компьютерного зрения могут выиграть от интеграции зрения и языка. Тем не менее одна задача, которая идеально подходит для изучения их интеграции, – это *обучение без ознакомления* (zero shot learning, ZSL, иногда называемое обучением без подготовки) – потому что, в отличие от других задач, ее нельзя решить без введения невизуальных знаний, например отраженных в языке.

ZSL – это задача с двумя наборами данных: обучающим и тестовым. Категории этих наборов разбиваются на «знакомые» и «незнакомые». Обучающий набор состоит только из «видимых» категорий, в то время как тестовый набор содержит «незнакомые» категории, а также, опционально, «знакомые». Например, может быть задача ZSL, которая включает знакомые категории «бегать» и «стоять», а также категорию «ходить», с которой модель не знакомилась при обучении. Задача обучения будет заключаться в том, чтобы модель научилась визуально распознавать и правильно классифицировать ходьбу, хотя она ей ранее никогда не встречалась, но категории «бег» и «стояние» встречались в обучающем наборе.

Существует несколько подходов к ZSL. Ранние работы по ZSL сосредоточены на *атрибутах* – визуально распознаваемых характеристиках с дифференциальными ассоциациями классов (например, классов объектов или классов действий). Атрибуты можно обобщить в том смысле, что детекторы атрибутов, обученные только на видимом наборе, способны обнаруживать атрибуты в выборках как из знакомого, так и из незнакомого набора.

Более поздние работы над ZSL, как правило, сосредоточены на *семантических пространствах представления*. Это евклидовы векторные пространства, в которых семантические категории (например, отраженные в языке) связаны с векторами или точками в пространстве. Эти векторные представления слов имеют значительно более низкую размерность, чем *прямое унитарное кодирование* с одним активным состоянием (one-hot encoding, векторы, в которых каждое измерение соответствует классу, а все измерения, кроме одного, равны нулю), и обладают таким свойством, что слова, сходные по семантике, представлены векторами, которые расположены рядом в пространстве представления.

В семантических пространствах представления символические представления векторизуются таким образом, что сохраняются семантические отношения категорий символов. Это позволяет интегрировать символическую семантику в глубокие архитектуры, внутренние представления которых состоят из векторов. Интеграция сводится к правильному сопоставлению ви-

зуальных векторных представлений с векторизованными символическими представлениями.

В разных методах ZSL используются разные общие представления: одни встраивают визуальные признаки в семантическое пространство, другие встраивают семантическое пространство в пространство визуальных признаков, а некоторые встраивают и то, и другое в третье общее пространство. Поскольку и визуальные, и семантические представления лежат в одном и том же пространстве, категоризация визуального ввода сводится к поиску ближайшей семантической метки в этом пространстве.

Современные методы ZSL часто используют CNN для визуальных признаков и создают общие пространства представлений с предварительно обученными семантическими пространствами представлений, такими как word2vec (Mandal et al., 2019; Xian et al., 2018). Некоторые методы ZSL, основанные на общем представлении, структурируют и обучают свои модели сквозным способом, что сложнее, но обеспечивает выигрыш в производительности (Zhang et al., 2017).

Оставшаяся часть текущего раздела 11.4 построена следующим образом: в разделе 11.4.1 мы подробно описываем простое использование атрибутов для ZSL и определение относительного атрибута с более тонкими нюансами; в разделе 11.4.2 представлено использование общих семантических пространств в ZSL; в разделе 11.4.3 мы рассмотрим основные подходы к построению семантического векторного пространства; в разделе 11.4.4 пойдет речь о включении знаний в виде графов для классификации действий по методу ZSL.

11.4.1. Атрибуты в обучении без ознакомления

Использование для распознавания деятельности атрибутов, в том числе атрибутов, ориентированных на действия, таких как аффордансы, описанные в разделе 11.2, позволяет создавать классификаторы, интерпретируемые и определяемые человеком. Атрибуты смягчают проблему непрозрачности модели за счет использования явного предопределенного представления среднего уровня ниже уровня категорий классов.

Использование атрибутов делает возможным простой механизм, с помощью которого можно изучать визуальные представления из доступных обучающих данных и передавать эти представления в классы, для которых нет доступных обучающих данных. Атрибуты являются общими в том смысле, что они присутствуют в нескольких категориях классов, а использование нескольких атрибутов с различным распределением по классам позволяет представлять конкретные классы.

Атрибуты в ZSL применяются следующим образом: детекторы атрибутов обучаются по знакомому набору и связанным с ним меткам атрибутов. Эти детекторы можно обобщить как на знакомый, так и на незнакомый набор. Из-за того, что разные категории атрибутов обладают разным охватом классов (например, класса объектов или действий), разные комбинации атрибутов могут представлять разные классы. Следовательно, детекторы клас-

сов могут быть расположены поверх детекторов атрибутов. Эта реализация опирается на спецификацию того, какие атрибуты связаны с классами. При наличии спецификации для незнакомых категорий можно построить соответствующие детекторы, даже если визуальные образцы незнакомых категорий не встречались.

Традиционное использование атрибутов в компьютерном зрении является бинарным: атрибут либо присутствует, либо отсутствует. Это ограничивает репрезентативную силу представлений атрибутов. Однако бинарные представления можно обобщить до скалярных представлений, где каждый атрибут связан со скалярной степенью, а не с бинарной категорией. Это более гибко с точки зрения представления и позволяет использовать атрибуты, которые не так четко подпадают под бинарную категоризацию. Например, в то время как атрибут «в помещении / на улице» обычно явно бинарный, атрибут «двигаться быстро/медленно» имеет более равномерное распределение по градациям визуального ввода.

Парих и Грауман (Parikh, Grauman, 2011) предложили один из подходов к обобщению бинарных атрибутов на скалярные. Проблемой при обобщении бинарных атрибутов на скалярные является неоднородность и непоследовательность аннотаций, поскольку разные аннотаторы могут по-разному толковать, чему соответствуют разные степени атрибутов в скалярном представлении. Некоторые исследователи решают эту проблему, требуя, чтобы аннотаторы не присваивали скалярные значения непосредственно атрибутам, а ранжировали изображения с точки зрения степени атрибута. После ранжирования изображений значения скалярных атрибутов могут быть получены из их аннотированных относительных степеней.

В случае бинарных атрибутов детекторы можно обучить с помощью обычных классификаторов, но для создания скалярных атрибутов требуются другие методы. Парих и Грауман (Parikh, Grauman, 2011) обучают функцию ранжирования изображений на атрибутах знакомого набора и используют эту функцию ранжирования для получения скалярного значения.

Относительные представления атрибутов обеспечивают большую гибкость в спецификациях классов. Используя атрибут «быстрое движение / медленное движение», можно указать, что незнакомая категория «бег» быстрее, чем знакомая категория «ходьба», или что незнакомая категория «стояние» медленнее, чем знакомая категория «ходьба». Это делается без необходимости определять бинарную спецификацию или интуитивно понятное скалярное значение для описания незнакомых категорий.

11.4.2. Общие пространства для встраивания

Обобщение бинарных атрибутов до скалярных увеличивает их репрезентативную силу. Однако увеличение репрезентативной способности произошло за счет увеличения сложности аннотирования атрибутов и определения классов. Одним из решений этой проблемы являются относительные атрибуты (Parikh, Grauman, 2011).

Атрибуты также могут быть абстрактными. Вовсе не обязательно, чтобы визуальные представления сопровождалось понятными человеку атрибутами. Абстрагирование атрибутов имеет два преимущества:

- отсутствие неточностей, возникающих при использовании сконструированных, а не изученных представлений;
- отсутствие накладных расходов и ошибок при аннотировании атрибутов.

Но тогда возникает вопрос, как построить систему, чтобы изучение классификации знакомых классов давало нам классификатор, способный обрабатывать не только знакомые, но и незнакомые классы без использования извлеченного из данных представления промежуточного уровня, которое позволяет специфицировать незнакомые классы с точки зрения этого представления.

Один из подходов состоит в том, чтобы определить незнакомые классы с точки зрения их отношений сходства со знакомыми классами, изучая отношения из корпусов текстов. В области обработки естественного языка (NLP) ведется обширная работа по созданию семантических векторных пространств, которые представляют отношения подобия между словами, – одним из популярных примеров является word2vec (Mikolov et al., 2013). Идея заключается в том, что термины в этих пространствах расположены в непосредственной близости от других терминов, с которыми они имеют семантическое сходство. Обученные семантические пространства общедоступны – их можно взять и использовать без необходимости создавать с нуля. В разделе 11.4.3 более подробно рассказано о создании таких пространств.

Термины с семантическим сходством часто также имеют сходство в визуальном пространстве – например, «бег трусцой» как семантически, так и визуально находится между «бегом» и «ходьбой». Часто бывает так, что если можно определить визуальное сходство с известными категориями, то можно установить и семантические отношения сходства. Затем из этих семантических отношений мы можем вывести семантические категории визуальной информации.

В качестве иллюстрации рассмотрим простой пример. Возьмем набор знакомых классов, включающий «бег» и «ходьбу», и набор незнакомых классов, включающий «бег трусцой», семантическое векторное пространство, отражающее семантическую близость между этими категориями, и архитектуру компьютерного зрения (например, CNN), которая создает визуальные представления входных данных. Пусть у нас есть визуальные входные данные, относящиеся к незнакомому классу. Внутреннее представление этого входа, созданное CNN, находится на полпути между представлением «бега» и «ходьбы». Затем мы переходим в семантическое языковое пространство и видим, что метка, расположенная на полпути между «бег» и «ходьба», – это «бег трусцой», и присваиваем эту метку входным данным.

Чаще всего сравнение сходства между образцами незнакомой класса и знакомыми классами производится не в визуальном пространстве. Обычно входные данные сопоставляются с семантическим пространством и там выполняются сравнения с известными классами. Это требует встраивания одного пространства в другое.

Механизм встраивания одного пространства в другое может быть таким же простым, как линейное преобразование, применяемое к одному пространству, которое обучается на потере сходства между двумя пространствами. Сеть DeVise (Frome et al., 2013) – хороший пример архитектуры, использующей этот метод. Она включает в себя два предварительно обученных представления – визуальные признаки, взятые, например, из CNN, обученной классификации, и векторы слов в пространстве представлений, которые могут быть созданы с помощью средств, обсуждаемых в разделе 11.4.3.

Один из методов встраивания визуальных признаков в семантическое пространство векторов слов показан на рис. 11.8. Слой узлов добавляется к верхней части предварительно обученных признаков CNN, а затем обучается. Потерей, которая обучает этот слой, является сходство (например, косинусное сходство) между выходными данными этого слоя и векторами в пространстве семантического представления, соответствующими меткам визуального ввода. Таким образом обучается функция простого линейного отображения векторов визуальных признаков на семантические признаки, полученные из текста.

Часто используются более сложные отображения, чем линейные. Использование нескольких слоев нейронов в сочетании с нелинейными активациями дает нелинейные отображения (например, Kato et al., 2018 используют такой метод). Кодиров и др. (Kodirov et al., 2017) для создания представления использовали автоэнкодер с семантическими ограничениями и обнаружили, что ограничение визуальной реконструкции приводит к лучшему обобщению незнакомых классов в ZSL.

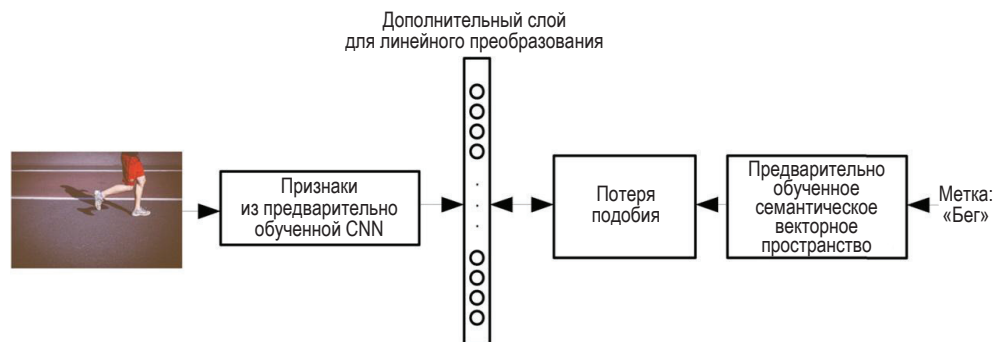


Рис. 11.8 ❖ Простой подход к встраиванию визуальных представлений в семантическое пространство, применяемый такими методами, как DeVise (Frome et al., 2013). Используются две предварительно обученные модели: 1) предварительно обученная модель извлечения визуальных признаков, такая как CNN, и 2) семантическое векторное пространство, созданное с помощью методов, обсуждаемых в разделе 11.4.3. Простое линейное преобразование производится путем добавления слоя узлов поверх визуальных признаков и их обучения с учетом меры их сходства с семантическим вектором слов меток, соответствующих зрительному вводу. Готовое линейное преобразование представляет собой отображение пространства визуальных признаков в пространство семантических векторов

Как только визуальный ввод и семантическое представление помещены в одно и то же пространство, определить класс нового визуального ввода так же просто, как представить визуальный ввод в этом пространстве, а затем найти ближайшую семантическую метку в том же пространстве.

11.4.3. Построение семантических векторных пространств

11.4.3.1. *word2vec*

Допустим, мы хотим построить векторное пространство, в котором слова представлены таким образом, чтобы их семантика определяла пространственное расположение вектора. Конечным результатом является пространство, в котором, например, векторы «бег», «трусца» и «ходьба» расположены рядом и «трусца» находится между «бегом» и «ходьбой».

Как определить семантику слова? Один из ответов заключается во взаимодействии слова с другими словами в текстовом корпусе – семантика слов может быть определена по отношению к другим словам. Как мы моделируем отношения слов к другим словам? Одним из простых подходов является совпадение: если два слова встречаются рядом, они считаются связанными. Чем чаще они встречаются вместе, тем сильнее они связаны. Это принцип, на котором основаны методы представления слов в векторных пространствах, такие как *word2vec*.

Начнем с простейшего векторного представления слов – прямого унитарного кодирования, где каждая позиция вектора соответствует одному слову в словаре V . Это многомерное неэффективное представление, в котором нет значимых пространственных отношений между словами. Мы можем разместить слова в пространстве меньшей размерности с указанными желаемыми свойствами, решив одну из двух связанных задач:

- 1) прогнозирование целевого слова на основе контекста;
- 2) прогнозирование контекста на основе целевого слова.

Здесь контекст C определяется как набор терминов «рядом» с целевым термином. Они определяются как другие термины, присутствующие в n -грамме, связанной с целевым термином, без учета порядка слов.

Каждая из этих задач может быть решена с использованием простых архитектур, и в процессе решения этих задач возникают представления меньшей размерности, которые затем можно взять и использовать для других задач.

Метод *Continuous Bag Of Words* (CBOW) для решения задачи 1 (Mikolov et al., 2013) показан на рис. 11.9а. Входные данные состоят из нескольких слов контекста, каждое из которых представлено как унитарный вектор размерности $|V|$. Эти векторы суммируются для получения вектора размером $|V|$. Он проходит через один слой из N нейронов, где N – размер пространства представления. Вдобавок к этому у нас есть еще один слой размером $|V|$, чья задача заключается в том, чтобы предсказать в унитарном коде слово, связанное с контекстом, состоящим из терминов, подаваемых в качестве входных данных для первого слоя.

Метод *Skip Gram* для решения задачи 2 (Mikolov et al., 2013) показан на рис. 11.9b. Вход состоит из одного термина, представленного как унитарный вектор размерности $|V|$. Он подается в один слой размерности N , где N – размерность пространства представления. Выше этого слоя у нас есть выходной слой, состоящий из $|C|$ наборов $|V|$ узлов, каждый из которых связан с одним термином контекста C в n -грамме, связанной с входным термином.

Обе архитектуры обучаются путем последовательной подачи n -грамм, извлеченных из больших текстовых корпусов, и применения потерь softmax к последнему слою для обучения соответствию ожидаемым терминам.

После того как эти архитектуры обучены, скрытый слой затем переносит отображение V из унитарного представления размера $|V|$ во встроенное представление размера N , где термины пространственно расположены рядом с терминами с аналогичной семантикой.

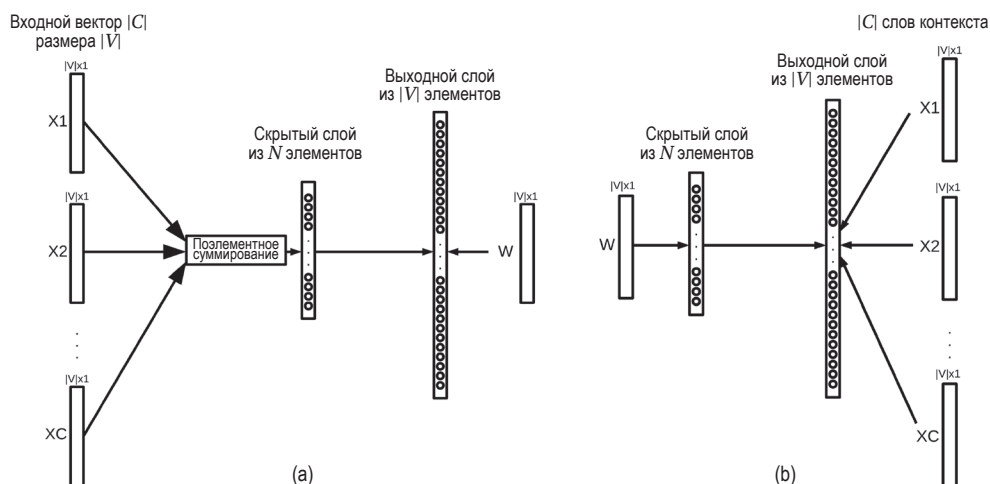


Рис. 11.9 ❖ (a) Метод Continuous Bag Of Words для решения задачи предсказания целевого слова на основе контекста этого слова (Mikolov et al., 2013). Входные векторы $|C|$ контекста проходят через два слоя, первый – скрытый слой размера N , второй – выходной слой размера V . Потеря рассчитывается относительно слова W . После обучения выходной слой можно отбросить, а скрытый слой использовать для перевода из унитарных кодировок слов во вложенное пространство размера N . (b) Метод Skip Gram для решения задачи предсказания контекста термина (Mikolov et al., 2013). Слово W проходит через два сетевых слоя, первый – скрытый слой размера N , второй – выходной слой размера V . Потеря рассчитывается относительно $|C|$ слов контекста. После обучения выходной слой можно отбросить, а скрытый слой использовать для перевода из унитарных кодировок слов в пространство представления размера N .

11.4.4. Общие пространства представления и графовые модели

Кроме простого отображения в общее пространство встраивания, ZSL действия может получить еще одну выгоду от дополнительной структуры, по-

сколько ее можно представить в виде графов. В нескольких работах (Ghosh et al., 2020a,b; Kato et al., 2018; Yan et al., 2018) для распознавания действий используются графы, которые обрабатываются с помощью *графовой сверточной сети* (graph convolutional network, GCN) для получения векторных представлений категорий действий, подходящих для ZSL.

Гош и др. (Ghosh et al., 2020) оценивают три различных графовых представления, последнее из которых применимо к обучению с *ограниченным ознакомлением* (few-shot learning), а не к ZSL, поскольку в нем используются визуальные признаки из небольшого количества образцов. Като и др. (Kato et al., 2018) строят граф на основе триплетов Subject Verb Object, полученных из корпусов знаний. Все они обрабатываются через GCN.

Во всех новых исследованиях (Ghosh et al., 2020) применяется sentence2vec (Pagliardini et al., 2017), а не word2vec, поскольку авторы считают, что категории действий лучше представлены фразами, чем отдельными словами, которые могут иметь несколько значений. Первый граф состоит из узлов, принимающих значения представлений sentence2vec для фраз, обозначающих категории действия. Эти узлы связаны друг с другом на основе косинусного подобия между векторными представлениями – N ближайших соседей для каждого узла соединены ребрами. Второй граф связывает глаголы и существительные, извлеченные из тегов Part-of-Speech (части речи) фраз, описывающих действия, с каждым классом действий, включая существительные в качестве прочной связи между знакомыми и незнакомыми категориями. Третий граф включает в себя визуальные представления, полученные из небольшого количества образцов незнакомых категорий, – поэтому мы говорим, что этот граф применим к обучению с ограниченным знакомством, а не к ZSL. Причина добавления визуальных репрезентаций заключается в том, что категории, которые сходны в семантическом пространстве, могут тем не менее иметь различные визуальные проявления – авторы приводят пример «выгула коня» и «конной прогулки», которые имеют схожие представления слов, но разные визуальные проявления.

Като и др. (Kato et al., 2018) строят граф, состоящий из трех типов узлов: узлов существительных, узлов объектов и узлов действий. Узлы действий связаны с глаголами и существительными, которые включают в себя связанное действие.

Узлы графа обычно инициализируют значениями, полученными из семантических векторных пространств – это важно, поскольку они задают начальные отношения, по которым итерируется GCN. Эта итерация охватывает отношения, определяемые ребрами в графе, и позволяет передавать информацию от узла к узлу вдоль ребер. Например, в (Kato et al., 2018) узлы действия, для которых изначально заданы нулевые векторы, приобретают представление, определяемое векторами ближайших узлов существительного и глагола, которые были инициализированы семантическими векторами.

Как и ранее упомянутые, эти методы изучают отображение визуальных признаков (взятых из CNN, предварительно обученной для отдельной задачи) в общее пространство представлений, хотя здесь это пространство используется совместно с представлениями, созданными GCN. У Като и др. (Kato et al., 2018) это отображение производится через два слоя нейронов

с сигмовидной активацией, что приводит к нелинейному отображению визуальных признаков в общее семантическое пространство. Как и в предыдущей работе, эти слои обучаются путем применения потери, измеряющей сходство между векторами визуальных признаков и векторами, ассоциированными с метками входных данных, созданными GCN.

Затем, используя знакомый обучающий набор, можно изучить отображение предварительно обученных визуальных признаков на векторы действия, созданные GCN. Чтобы получить прогнозы сети для новых действий во время тестирования, можно применить алгоритм поиска ближайших соседей между визуальными признаками и векторами действия.

11.5. ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ

В этом разделе мы обсудим следствия, вытекающие из применения ориентированной на действия структуры сети и перспективы дальнейших исследований. От действия зависят *задачи и наборы данных* – оно позволяет проводить концептуальное моделирование, способствующее обобщению и долгосрочному временному прогнозированию. Пониманию действия способствует моделирование *концепций*, выходящих за рамки обычного компьютерного зрения. По мере развития компьютерного зрения масштабируемость традиционных методов обучения с учителем становится все более серьезной проблемой, и методы, основанные на действиях, помогают смягчить эту проблему, используя преимущества парадигм обучения с частичным привлечением учителя или совсем без учителя. Наконец, действие помогает интегрировать познание и символическое моделирование в восприятие, в том числе в нескольких модальностях восприятия.

Задачи и наборы данных. Деятельность охватывает длительные промежутки времени. Следовательно, при поиске решений для понимания наблюдаемой деятельности мы сталкиваемся с проблемами гораздо более сложными, чем те, с которыми мы сталкиваемся в текущих задачах распознавания действий. Объекты, действия, аффордансы и другие составляющие сцены семантически соотносятся друг с другом в различных временных масштабах, и нам необходимо найти способы моделирования этих отношений. Мы думаем, что эта способность не очень хорошо раскрыта в задаче распознавания действий. Нам следует выбирать задачи, которые демонстрируют обобщение и концептуальное понимание действия (в отличие от понимания, основанного исключительно на внешнем виде). Такие задачи включают обучение без ознакомления и предсказание будущих действий, а также переход с одной точки зрения на другую (например, от первого лица к третьему лицу). Сегодняшние исследования компьютерного зрения в значительной степени обусловлены появлением новых наборов данных и определением новых задач – существующие наборы данных недостаточно охватывают долгосрочное и концептуальное моделирование деятельности. Наборы данных в идеале должны иметь записи с разных точек зрения, потому что это открывает

возможности для интересных исследований, например для решения задачи передачи знаний между видом от первого и третьего лица. Наконец, большинство наборов данных содержат сцены в помещении. Будет интересно собрать набор сцен на открытом воздухе и проанализировать их, как обсуждалось выше, рассматривая связи между аффордансами, взаимодействиями и долгосрочными отношениями. Мы также могли бы попытаться провести основанный на действиях анализ данных из области автономного вождения.

Концепции понимания длительной деятельности. Понимание деятельности требует наличия четкой концепции анализа и распознавания действий в различных временных масштабах. В этой главе мы обсуждали элементы такой концепции применительно к одиночному изображению и в краткосрочных масштабах времени, включая аффордансы, захваты рук, геометрические отношения. Мы отмечали важность использования процессов геометрической реконструкции из-за их робастности (раздел 11.3.4). Следующим шагом будет добавление дополнительных ограничений робастности для моделирования временных отношений в более длинных промежутках времени. Мы могли бы использовать онтологии для классификации объектов и действий. Сегодня мы можем классифицировать глаголы, исходя из эффектов действия (Yang et al., 2013), принципов эргономики или ограничений, связанных с силой и местоположением. Долгосрочные отношения охватывают причинно-следственные связи и ограничения на возможные и невозможные последовательности действий. Мы также можем моделировать физические ограничения и использовать физические движки, но для того, чтобы интегрировать их в глубокие архитектуры, нам нужно встроить эти ограничения в векторные пространства, которые связывают восприятие с познанием (раздел 11.4).

Уменьшение потребности в обучении с учителем. Ранние подходы к интеграции языка со зрением (разделы 11.4.1 и 11.4.4) в значительной степени были основаны на обучении с учителем. Например, распознавание визуальных атрибутов или объектов было реализовано посредством обучения с учителем. Модели на основе графов, использующие общие представления для ZSL, должны быть заранее готовы распознать категории «незнакомых» набора, с которыми они могут столкнуться во время применения. Естественно, в системе, основанной на действиях, найдут воплощение различные варианты концепции развития, такие как обучение без учителя и самообучение, перенос знаний, метаобучение и в конечном итоге *бесконечное обучение* (Mitchell et al., 2015). Например, при построении визуальных онтологий нам не следует полностью полагаться на обучение с учителем при изучении визуальных представлений классов метаклагов. Одним из способов такого моделирования является изучение словаря (Zheng et al., 2016), но до сих пор эти подходы ограничивались простыми действиями. Нам потребуются методы, которые масштабируются на более сложные действия, выполняемые людьми. Генеративно-состязательные сети (GAN) и вариационные автокодировщики (VAE) продемонстрировали свою эффективность в моделировании изображений и видео с атомарными действиями. Мы могли бы, например, использовать VAE для изучения базового распределения данных в дискретном скрытом пространстве, которое кодирует метакатегории.

Многомодальное восприятие. Человеческое понимание мира основано на нашем двигательном познании и всех наших чувствах. Точно так же наши модели должны включать другие сенсорные модальности, такие как слуховые, тактильные или проприоцептивные¹ сигналы, в дополнение к моделированию зрительных и когнитивных представлений. Различные модальности предоставляют разную дополнительную информацию и поэтому допускают различные способы организации понятий. В дополнение к вопросам обучения с использованием различных модальностей нам также нужны методы доступа к сохраненным понятиям при восприятии из любой модальности. При изучении интеграции различных модальностей также было бы полезно глубже вникнуть в механизмы памяти. В качестве теоретической модели искусственного интеллекта был предложен фреймворк, известный как *векторные символические архитектуры* (vector symbolic architectures, VSA) (Plate, 1995; Eliasmith, 2013), включающий в себя гиперпространственные вычисления (методы, использующие векторные пространства очень высокой размерности) (Pentti, 2009). Гиперпространственные вычисления сочетают в себе преимущества подходов искусственного интеллекта на основе нейронных сетей с систематической композиционностью и упорядоченным поведением из области классического символического искусственного интеллекта (Levy, Gayler, 2008). В этом фреймворке концепции (идеи) кодируются и переносятся в векторные пространства, а алгебраические операции определяются в векторных пространствах. Эти операции представляют собой добавление родственных понятий и связывание векторов разного происхождения – например, это может быть звук и зрение или зрение и моторика (Mitrokhin et al., 2019). Операции поддерживают отделимость одной модальности от другой. Мы могли бы взять этот фреймворк за основу и объединить его с подходами нейронных сетей. Цель будет состоять в том, чтобы сохранить возможность кодировать в память явные модальности восприятия, сохраняя при этом способность вспоминать информацию из любой модальности.

11.6. Выводы

Целью компьютерного зрения является создание интерпретаций, полезных для людей. Действие занимает центральное место в нашем понимании мира, но недостаточно используется в современном компьютерном зрении. Эта глава посвящена пониманию сцены и деятельности, в основе которого лежит концепция действия и взаимодействия. Мы рассмотрели основанные на действиях подходы к пониманию сцены, в том числе моделирование в нескольких временных масштабах, начиная с интерпретации объекта с точки зрения аффордансов на уровне текущего момента и переходя к базовым действиям, а затем и к деятельности в более длительном временном масштабе. Мы

¹ *Проприоцепция* – это ощущение особого рода, возникающее в результате обработки сигналов от специализированных рецепторов (проприоцепторов), дающих мозгу информацию о положении мышц, сухожилий и суставов и в конечном счете о положении тела и его частей в пространстве. – *Прим. перев.*

описали хорошо развитую область аффордансного обучения и представили краткий обзор работ по пониманию деятельности, сочетающих когнитивный и лингвистический подходы с интерпретируемыми людьми модулями, необходимыми для характеристики деятельности и временной сегментации видео. Мы обсудили методы интеграции визуальных представлений со знаниями, как созданными, так и полученными из текстовых корпусов. В главе была рассмотрена интеграция видения, сосредоточенного на действии, с графовыми методами и представлены возможные направления исследований, включая создание новых задач и наборов данных, развитие концепций кодирования долгосрочных отношений, применение методов обучения без учителя и добавление памяти в качестве центрального компонента системы, предназначенной для понимания деятельности.

БЛАГОДАРНОСТИ

Благодарим за поддержку исследований по грантам BCS 1824198 и OISE 2020624 со стороны Национального фонда науки.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Aditya Somak, Yang Yezhou, Baral Chitta, Aloimonos Yiannis, Fermüller Cornelia*, 2018. Image understanding using vision and reasoning through scene description graph. *Computer Vision and Image Understanding* 173, 33–45.
- Aloimonos Yiannis, Fermüller Cornelia*, 2015. The cognitive dialogue: a new model for vision implementing common sense reasoning. *Image and Vision Computing* 34, 42–44.
- Ansuini Caterina, Cavallo Andrea, Bertone Cesare, Becchio Cristina*, 2015. Intentions in the brain: the unveiling of mister Hyde. *The Neuroscientist* 21 (2), 126–135.
- Bajcsy Ruzena*, 1988. Active perception. *Proceedings of the IEEE* 76 (8), 966–1005.
- Barsalou Lawrence W.*, 2008. Grounded cognition. *Annual Review of Psychology* 59, 617–645.
- Bo Liefeng, Ren Xiaofeng, Fox Dieter*, 2013. Unsupervised feature learning for rgb-d based object recognition. In: *Experimental Robotics*. Springer, pp. 387–402.
- Cutkosky Mark R.*, 1989. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation* 5 (3), 269–279.
- Damen Dima, Doughty Hazel, Maria Farinella Giovanni, Fidler Sanja, Furnari Antonino, Kazakos Evangelos, Moltisanti Davide, Munro Jonathan, Perrett Toby, Price Will, et al.*, 2018. Scaling egocentric vision: the EPICkitchens dataset. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–736.
- Das Pradipto, Xu Chenliang, Doell Richard F., Corso Jason J.*, 2013. A thousand frames in just a few words: lingual description of videos through latent topics and sparse object stitching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2634–2641.

- Dessalene Eadom, Devaraj Chinmaya, Maynord Michael, Fermüller Cornelia, Aloimonos Yiannis*, 2021. Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Dutta V., Zielinska T.*, 2017. Action prediction based on physically grounded object affordances in human-object interactions. In: *Proceedings of the 11th International Workshop on Robot Motion and Control*.
- Ecins Aleksandrs, Fermüller Cornelia, Aloimonos Yiannis*, 2016. Cluttered scene segmentation using the symmetry constraint. In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2271–2278.
- Eliasmith Chris*, 2013. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press.
- Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A.*, 2010. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88 (2), 303–338.
- Fermüller Cornelia, Aloimonos Yiannis*, 1995. Vision and action. *Image and Vision Computing* 13 (10), 725–744.
- Fermüller Cornelia, Wang Fang, Yang Yezhou, Zampogiannis Konstantinos, Zhang Yi, Barranco Francisco, Pfeiffer Michael*, 2018. Prediction of manipulation actions. *International Journal of Computer Vision* 126 (2), 358–374.
- Fitzpatrick Paul, Metta Giorgio, Natale Lorenzo, Rao Sajit, Sandini Giulio*, 2003. Learning about objects through action-initial steps towards artificial cognition. In: *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 3140–3145.
- Frome Andrea, Corrado Greg, Shlens Jonathon, Bengio Samy, Dean Jeffrey, Ranzato Marc'Aurelio, Devise Tomas Mikolov*, 2013. A deep visual-semantic embedding model. *Advances in Neural Information Processing Systems* 26.
- Ghosh Pallabi, Saini Nirat, Davis Larry S., Shrivastava Abhinav*, 2020a. All about knowledge graphs for actions. *arXiv preprint. arXiv:2008.12432*.
- Ghosh Pallabi, Yao Yi, Davis Larry, Divakaran Ajay*, 2020b. Stacked spatio-temporal graph convolutional networks for action segmentation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 576–585.
- Gibson James J.*, 1977. The theory of affordances. In: *Bransford John, Shaw Robert E.* (Eds.), *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 67–82.
- Grabner Helmut, Gall Jürgen, Van Gool Luc*, 2011. What makes a chair a chair? In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1529–1536.
- Guha Anupam, Yang Yezhou, Fermüller Cornelia, Aloimonos Yiannis*, 2013. Minimalist plans for interpreting manipulation actions. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5908–5914.
- Gupta Abhinav, Satkin Scott, Efros Alexei A., Hebert Martial*, 2011. From 3d scene geometry to human workspace. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1961–1968.
- Hassanin Mohammed, Khan Salman, Tahtali Murat*, 2018. Visual affordance and function understanding: a survey. *arXiv preprint. arXiv:1807.06775*.
- Hedau Varsha, Hoiem Derek, Forsyth David*, 2009. Recovering the spatial layout of cluttered rooms. In: *IEEE International Conference on Computer Vision*, pp. 1849–1856.

- Hermans Tucker, Reh James M., Bobick Aaron*, 2011. Affordance prediction via learned object attributes. In: IEEE International Conference on Robotics and Automation (ICRA): Workshop on Semantic Perception, Mapping, and Exploration.
- Pentti Kanerva*, 2009. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation* 1, 139–159.
- Kato Keizo, Li Yin, Gupta Abhinav*, 2018. Compositional learning for human object interaction. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 234–251.
- Kjellström Hedvig, Romero Javier, Kragić Danica*, 2011. Visual object-action recognition: inferring object affordances from human demonstration. *Computer Vision and Image Understanding* 115 (1), 81–90.
- Kodirov Elyor, Xiang Tao, Gong Shaogang*, 2017. Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3174–3183.
- Koppula Hema S., Saxena Ashutosh*, 2014. Physically grounded spatio-temporal object affordances. In: European Conference on Computer Vision. Springer, pp. 831–847.
- Koppula Hema S., Saxena Ashutosh*, 2015. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (1), 14–29.
- Koppula Hema Swetha, Gupta Rudhir, Saxena Ashutosh*, 2013. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research* 32 (8), 951–970.
- Lee David C., Gupta Abhinav, Hebert Martial, Kanade Takeo*, 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In: Advances in Neural Information Processing Systems, pp. 1288–1296.
- Levy S. D., Gayler R.*, 2008. Vector symbolic architectures: a new building material for artificial general intelligence. In: Proceedings of the First Conference on Artificial General Intelligence (AGI-08). IOS Press.
- Liu Fayao, Shen Chunhua, Lin Guosheng*, 2015. Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162–5170.
- Mandal Devraj, Narayan Sanath, Kumar Dwivedi Sai, Gupta Vikram, Ahmed Shuaib, Shahbaz Khan Fahad, Shao Ling*, 2019. Out-of-distribution detection for generalized zero-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9985–9993.
- Martin A.*, 2007. The representation of object concepts in the brain. *Annual Review of Psychology* 58, 25–45.
- Metta Giorgio, Sandini Giulio, Vernon David, Natale Lorenzo, Nori Francesco*, 2008. The iCub humanoid robot: an open platform for research in embodied cognition. In: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 50–56.
- Mikolov Tomas, Chen Kai, Corrado Greg, Dean Jeffrey*, 2013. Efficient estimation of word representations in vector space. arXiv preprint. arXiv:1301.3781.

- Miller George A., Beckwith Richard, Fellbaum Christiane, Gross Derek, Miller Katherine J., 1990. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography* 3 (4), 235–244.
- Mitchell T., Cohen W., Hruschka E., Talukdar P., Betteridge J., Carlson A., Dalvi B., Gardner M., Kisiel B., Krishnamurthy J., Lao N., Mazaitis K., Mohamed T., Nakashole N., Platanios E., Ritter A., Samadi M., Settles B., Wang R., Wijaya D., Gupta A., Chen X., Saparov A., Greaves M., Welling J., 2015. Never-ending learning. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*.
- Mitrokhin A., Sutor P., Fermüller C., Aloimonos Y., 2019. Learning sensorimotor control with neuromorphic sensors: toward hyperdimensional active perception. *Science Robotics* 4 (30), eaaw6736.
- Montesano Luis, Lopes Manuel, Bernardino Alexandre, Santos-Victor José, 2008. Learning object affordances: from sensory–motor coordination to imitation. *IEEE Transactions on Robotics* 24 (1), 15–26.
- Myers Austin, Teo Ching L., Fermüller Cornelia, Aloimonos Yiannis, 2015. Affordance detection of tool parts from geometric features. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1374–1381.
- Nguyen Anh, Kanoulas Dimitrios, Caldwell Darwin G., Tsagarakis Nikos G., 2017. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5908–5915.
- Pagliardini Matteo, Gupta Prakhar, Jaggi Martin, 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint. arXiv:1703.02507*.
- Parikh Devi, Grauman Kristen, 2011. Relative attributes. In: *International Conference on Computer Vision*, pp. 503–510.
- Pastra Katerina, Aloimonos Yiannis, 2012. The minimalist grammar of action. *Philosophical Transactions of the Royal Society B: Biological Sciences* 367 (1585), 103–117.
- Plate Tony A., 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks* 6 (3), 623–641.
- Qi Siyuan, Huang Siyuan, Wei Ping, Zhu Song-Chun, 2017. Predicting human activities using stochastic grammar. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1164–1172.
- Qi Siyuan, Wang Wenguan, Jia Baoxiong, Shen Jianbing, Zhu Song-Chun, 2018. Learning human-object interactions by graph parsing neural networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 401–417.
- Roy Anirban, Todorovic Sinisa, 2016. A multi-scale cnn for affordance segmentation in rgb images. In: *European Conference on Computer Vision*. Springer, pp. 186–201.
- Russakovsky Olga, Deng Jia, Su Hao, Krause Jonathan, Satheesh Sanjeev, Ma Sean, Huang Zhiheng, Karpthy Andrej, Khosla Aditya, Bernstein Michael, et al., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (3), 211–252.

- Schuler Karin Kipper*, 2005. VerbNet: a broad-coverage, comprehensive verb lexicon. PhD thesis. Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA.
- Sigurdsson Gunnar A., Russakovsky Olga, Gupta Abhinav*, 2017. What actions are needed for understanding human actions in videos? In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2137–2146.
- Silberman Nathan, Hoiem Derek, Kohli Pushmeet, Fergus Rob*, 2012. Indoor segmentation and support inference from rgb-d images. In: European Conference on Computer Vision. Springer, pp. 746–760.
- Simonyan Karen, Zisserman Andrew*, 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556.
- Srikantha Abhilash, Gall Jürgen*, 2016. Weakly supervised learning of affordances. arXiv preprint. arXiv:1605.02964.
- Stark Louise, Bowyer Kevin*, 1991. Achieving generalized object recognition through reasoning about association of function to structure. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (10), 1097–1104.
- Summers-Stay Douglas, Teo Ching L., Yang Yezhou, Fermüller Cornelia, Aloimonos Yiannis*, 2012. Using a minimal action grammar for activity understanding in the real world. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4104–4111.
- Teo Ching Lik, Fermüller Cornelia, Aloimonos Yiannis*, 2015. Detection and segmentation of 2d curved reflection symmetric structures. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1644–1652.
- Tran Son D., Davis Larry S.*, 2008. Event modeling and recognition using Markov logic networks. In: European Conference on Computer Vision. Springer, pp. 610–623.
- Ugur Emre, Erhan Oztop, Erol Sahin*, 2011. Goal emulation and planning in perceptual space using learned affordances. Robotics and Autonomous Systems 59 (7–8), 580–595.
- Ugur Emre, Piater Justus*, 2016. Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection. IEEE Transactions on Cognitive and Developmental Systems 9 (4), 328–340.
- Varela Francisco J., Rosch Eleanor, Thompson Evan*, 1993. The Embodied Mind: Cognitive Science and Human Experience. MIT Press.
- Wörgötter Florentin, Erdal Aksoy Eren, Krüger Norbert, Piater Justus, Ude Ales, Tamosiunaite Minija*, 2013. A simple ontology of manipulation actions based on hand-object relations. IEEE Transactions on Autonomous Mental Development 5 (2), 117–134.
- Wörgötter Florentin, Ziaetabar F., Pfeiffer S., Kaya O., Kulvicius T., Tamosiunaite M.*, 2020. Humans predict action using grammar-like structures. Scientific Reports 10 (1), 1–11.
- Xian Yongqin, Lorenz Tobias, Schiele Bernt, Akata Zeynep*, 2018. Feature generating networks for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5542–5551.
- Xiao Jianxiong, Hays James, Ehinger Krista A., Oliva A., Torralba A.*, 2010. Sun database: large-scale scene recognition from abbey to zoo. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 3485–3492.

- Yan Sijie, Xiong Yuanjun, Lin Dahua*, 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32.
- Yang Yezhou, Fermüller Cornelia, Aloimonos Yiannis*, 2013. Detection of manipulation action consequences (MAC). In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2563–2570.
- Yang Yezhou, Guha Anupam, Fermüller Cornelia, Aloimonos Yiannis*, 2014. A cognitive system for understanding human manipulation actions. *Advances in Cognitive Systems* 3, 67–86.
- Yang Yezhou, Fermüller Cornelia, Li Yi, Aloimonos Yiannis*, 2015a. Grasp type revisited: a modern perspective on a classical feature for vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 400–408.
- Yang Yezhou, Li Yi, Fermüller Cornelia, Aloimonos Yiannis*, 2015b. Robot learning manipulation action plans by “watching” unconstrained videos from the world wide web. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29.
- Ye Chengxi, Yang Yezhou, Mao Ren, Fermüller Cornelia, Aloimonos Yiannis*, 2017. What can I do around here? Deep functional scene understanding for cognitive robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 4604–4611.
- Yu Xiaodong, Fermüller Cornelia, Teo Ching Lik, Yang Yezhou, Aloimonos Yiannis*, 2011. Active scene recognition with vision and language. In: 2011 International Conference on Computer Vision, pp. 810–817.
- Zampogiannis Konstantinos, Fermüller Cornelia, Cilantro Yiannis Aloimonos*, 2018. A lean, versatile, and efficient library for point cloud data processing. In: Proceedings of the 26th ACM International Conference on Multimedia, pp. 1364–1367.
- Zampogiannis Konstantinos, Fermüller Cornelia, Aloimonos Yiannis*, 2019. Topology-aware non-rigid point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zampogiannis Konstantinos, Yang Yezhou, Fermüller Cornelia, Aloimonos Yiannis*, 2015. Learning the spatial semantics of manipulation actions through preposition grounding. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1389–1396.
- Zhang Li, Xiang Tao, Gong Shaogang*, 2017. Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2021–2030.
- Zheng J., Jiang Z., Chellappa R.*, 2016. Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing* 25 (6), 2542–2556.

ОБ АВТОРАХ ГЛАВЫ

Корнелия Фермюллер – научный сотрудник Института перспективных компьютерных исследований Мэрилендского университета. Она получила докторскую степень в Венском технологическом университете и степень ма-

гистра в Технологическом университете Граца, обе по прикладной математике. Ее исследовательский интерес заключался в том, чтобы понять принципы систем активного зрения и разработать методы, основанные на подобию биологическим системам, особенно в области движения. Ее последняя работа была посвящена интерпретации действий человека и разработке алгоритмов трехмерного движения для экстремальных условий с использованием датчиков на основе событий.

Майкл Мейнорд – докторант кафедры компьютерных наук Колледж-Парка Университета Мэриленда, советниками которого являются Яннис Алоимонос и Корнелия Фермюллер. Область его исследовательских интересов охватывает символический искусственный интеллект, когнитивные архитектуры, компьютерное зрение, понимание действий и методы интеграции ИИ и компьютерного зрения.

Глава 12

.....

Сегментация событий во времени с использованием когнитивного самообучения

Авторы главы:

Рами Мунир, кафедра вычислительной техники и технологии,
Университет Южной Флориды, Тампа, Флорида, США;

Сатьянараянан Аакур, факультет информатики,
Государственный университет Оклахомы,
Стилуотер, Оклахома, США;

Судип Саркара

Краткое содержание главы:

- мы можем использовать теории когнитивной науки в области сегментации событий для разработки высокоэффективных алгоритмов компьютерного зрения, которые выполняют пространственно-временную сегментацию событий в видеопотоке, не требуя каких-либо размеченных заранее данных;
- мы обсудим три современные версии прогнозной модели восприятия: временная сегментация с использованием архитектуры перцептивного предсказания, временная сегментация с рабочими моделями событий, основанными на картах внимания, и, наконец, пространственная и временная локализация событий;
- вышеупомянутые современные методы могут изучить надежные представления событий всего лишь из одного прохода через немаркированное потоковое видео;
- новые методы демонстрируют уникальную точность в задаче временной сегментации и пространственно-временной локализации действий с обучением без учителя, предлагая конкурентоспособное качество по сравнению с базовыми моделями обучения с учителем, которые требуют большого объема аннотированных данных.

12.1. ВВЕДЕНИЕ

Как мы обнаруживаем и сегментируем события? Как мы представляем события? Как мы воспринимаем события? И что более важно, что такое событие? В исследованиях компьютерного зрения термины *действие* (action), *деятельность* (activity) и *событие* (event) часто смешивают. В большинстве работ эти термины используют попеременно, чтобы обозначить что-то, что происходит в сцене с участием объектов и действующих персон (*акторов*) и может быть аннотировано текстовой меткой, например «прыжки», «нарезание лука», «замена шин», «приготовление еды» и т. д. В существующей литературе по компьютерному зрению нет четкого различия между распознаванием действий, деятельностью и событий. Много неясного и в определениях характера событий. С другой стороны, восприятие событий является зрелой областью исследований когнитивной науки (Radvansky, Zacks, 2014; Shipley, Zacks, 2008; Richmond, Zacks, 2017). Мы начнем с обобщения некоторых достижений когнитивной науки, которые используем в качестве источника вдохновения для создания решений, изложенных в этой главе. В идеале следует прочитать первоисточники из приведенного в конце главы списка, а не полагаться только на наш обзор, который является лишь кратким введением в гораздо более богатую область знаний. Здесь мы выделим лишь некоторые идеи, которые использовали для создания решений компьютерного зрения в области временной сегментации событий с самообучением.

События являются ключевыми компонентами нашего опыта. Мозг получает непрерывный поток сенсорной информации как из внешнего мира, так и из тела, и сегментирует их на дискретные единицы или пакетные представления, называемые *событиями*. Каждое событие указывает на ключевые моменты входного сенсорного потока. Следовательно, событие определяется как *«отрезок времени в данном месте, который воспринимается наблюдателем как имеющий начало и конец»* (Zacks, Tversky, 2001). Обратите внимание, что это определение событий отличает их от деятельности. Например, приготовление пищи как таковое – это деятельность, а приготовление салата – это событие. У приготовления салата есть начало, середина и конец. События могут быть разных типов и продолжительности в зависимости от агента и воздействующей на него среды. Некоторые события короткие, например заправить постель. Некоторые события продолжительные, например матч по крикету. События с участием осмысленных агентов, таких как люди и животные, часто являются целенаправленными. Хотя цель этих событий не всегда сразу видна наблюдателю, она существует как задача, которая направляет событие. Существуют также события, в которых не участвуют осмысленные агенты и которые не имеют целей, например природные явления.

Широкой популярностью пользуется идея о том, что человеческое познание использует *«структурированные представления событий, называемые моделями событий, для сбора информации о пространственно-временной структуре, сущностях и объектах, а также других существенных характеристиках ситуации»* (Richmond, Zacks, 2017). Эти модели событий являются композиционным представлением события и составляющих его элементов и имеют *партономию*, т. е. иерархии, образованные отношением «часть–целое». На

рис. 12.1 показаны пример события «сделать бутерброд» и его иерархическая структура. На самом нижнем уровне иерархии находятся элементарные действия, такие как «принести нож» и «открыть крышку». Эти действия являются частью более длительных блоков действий, таких как «разрезать булочку», которые, в свою очередь, являются частью события «сделать бутерброд». Партономия действий аналогична партономии предметов, которые также могут быть описаны как композиция отдельных частей.

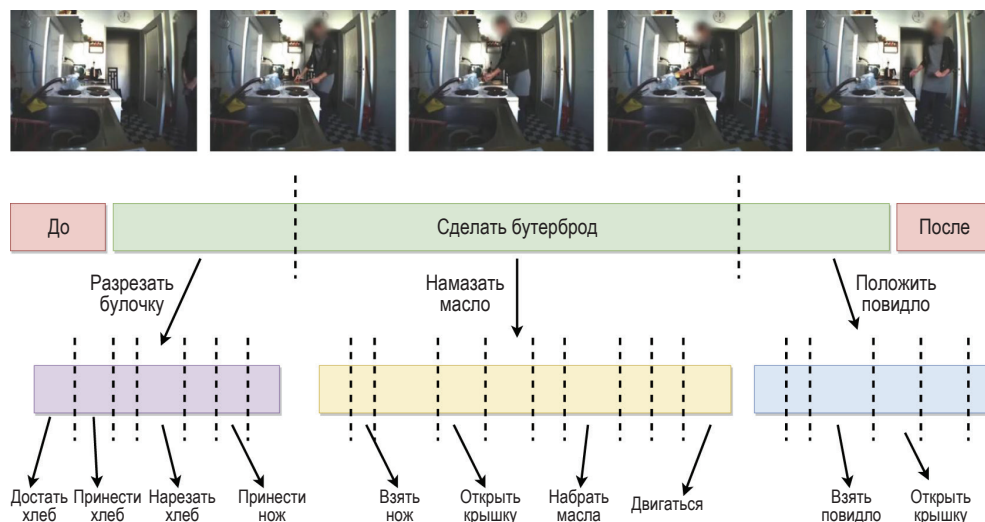


Рис. 12.1 ❖ Иерархия событий состоит из нескольких уровней сегментации. Событие более высокого уровня «Приготовление бутерброда» можно разделить на события более низкого уровня «Разрезание булочки», «Намазывание маслом» и «Добавление повидла». Каждое событие более низкого уровня может быть далее сегментировано на составляющие его события на еще более низком уровне. Событие «Сделать бутерброд» может быть частью события более высокого уровня «Позавтракать». Это композиционное отношение между событиями образует иерархию событий. Изображения взяты из набора данных «Действия за завтраком» (Kuehne et al., 2014)

Так же, как и части физических объектов, которые имеют видимые границы, события имеют границы сегментации в нескольких временных масштабах. Сегментация событий и группировка сегментов в иерархию – это непрерывные процессы, происходящие одновременно в нескольких временных масштабах. Данные нейробиологических экспериментов (Zacks et al., 2001a) указывают на то, что задняя височная область, теменная кора и латеральная лобная кора становятся активными во время достижения границ событий. Некоторые эксперименты (Kurby, Zacks, 2008) показывают, что сегментация событий играет важную роль в основных функциях когнитивного восприятия и кодирования памяти. Однако процесс сегментации событий не требует сознательного внимания. Сегментация событий может быть обработана исключительно восходящими признаками на основе сигналов, извлеченными из движения и внешнего вида. Этот вывод был основан на киноэксперимен-

те (Zacks et al., 2001; Zacks, Magliano, 2011). Некоторым участникам были показаны видеоролики о событиях, и их попросили отметить границы событий с помощью нажатия кнопки во время просмотра. Других участников попросили сделать то же самое, но им показали фильм наоборот! Границы, отмеченные обеими группами участников, были удивительно похожи. Изменение направления фильма не позволяет зрителям полагаться на знакомые схемы для интерпретации событий, что затрудняет идентификацию высокоуровневой информации, такой как достижение цели (Hard et al., 2006). Это свидетельствует в пользу предположения, что высокоуровневая информация о событии не является необходимой для сегментации событий. Следовательно, мы можем поставить перед собой задачу разработать системы компьютерного зрения, которым не нужны предварительные обучающие метки для сегментации событий, т. е. разработать самообучаемые алгоритмы сегментации событий в непрерывном режиме реального времени.

В этой главе мы рассматриваем проблему сегментации событий, т. е. то, каким образом мы отмечаем временные границы (когда одно событие заканчивается и начинается другое) и локализуем их в пространстве изображения. Построение событийной модели и партономической иерархии – это отдельный процесс, который мы здесь не рассматриваем. Сначала мы рассмотрим модель *теории сегментации событий* (event segmentation theory, EST) для вычисления границ событий (Zacks et al. (2014)), основанную на модели перцептивного предсказания. Затем представим наше решение для компьютерного зрения, основанное на модели перцептивного предсказания из EST, в трех последовательных версиях: временная сегментация с использованием структуры прогнозирования восприятия, временная сегментация вместе с рабочими моделями событий, основанными на картах внимания, и, наконец, пространственная и временная локализация событий. Мы закончим главу обсуждением других решений для сегментации событий, представленных в литературе, и того, как они соотносятся с нашими подходами.

12.2. ТЕОРИЯ СЕГМЕНТАЦИИ СОБЫТИЙ В КОГНИТИВНОЙ НАУКЕ

Теория сегментации событий (EST), разработанная Заком и др. (Zacks et al., 2007), основана на экспериментах по когнитивной нейробиологии. Она утверждает, что люди сохраняют стабильное представление о том, «что происходит сейчас», которое обновляется на основе проходящего увеличения *ошибки перцептивного предсказания* (perceptual prediction error). Это представление текущего события называется *моделью события* и используется для предсказания следующего сенсорного ввода. Этот процесс предсказания является ключевым элементом данной теории; прогнозы играют центральную роль в построении представления событий.

Наш мозг постоянно делает прогнозы того, с какими признаками окружающей среды наши органы чувств столкнутся в ближайшем будущем. Когда

мы наблюдаем, как кто-то готовит пищу, мы постоянно делаем прогнозы. Эти прогнозы делаются с разной степенью дискретности. Мы предсказываем траекторию движения в краткосрочном, практически мгновенном масштабе, чтобы предвидеть положение руки в следующий момент времени. И, в более грубом масштабе, пытаемся предвидеть, какая посуда будет использована следующей. Существует определение искусственного интеллекта, согласно которому способность делать прогнозы является ключевой характеристикой (Hawkins, Blakeslee, 2004). Степень, в которой агент может считаться разумным, определяется временным окном и пространственной областью, в которой он может делать точные прогнозы. Например, маленькое насекомое может предсказывать свое ближайшее окружение и свое ближайшее будущее. Люди, вооруженные научными рассуждениями и логикой высокого уровня, могут предсказывать события в гораздо большем пространстве и в более длительных временных окнах. Содержание прогноза зависит от поставленной задачи. В нашем случае речь идет о предсказании признаков визуального события.

Ошибка в прогнозе приводит к тому, что процесс сегментации группирует события в дискретные интервалы времени. Как показано на рис. 12.2, EST

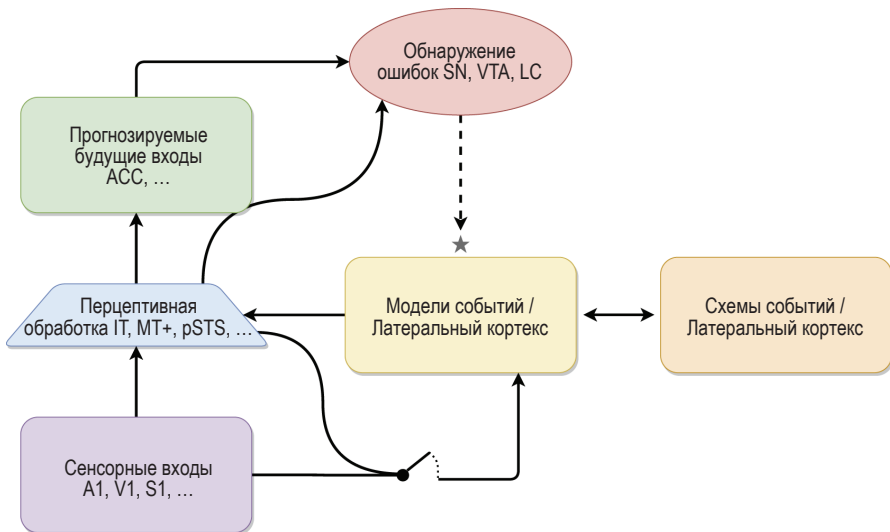


Рис. 12.2 ❖ Информационный поток в соответствии с теорией сегментации событий (EST), согласно исследованию (Zacks et al., 2007). Буквенные аббревиатуры относятся к различным областям мозга, в которых происходят эти действия, как это было обнаружено в экспериментах по когнитивной нейробиологии. Блок *перцептивной обработки* использует *сенсорные входы* для извлечения признаков более высокого уровня, имеющих отношение к прогнозирующей задаче. Извлеченные признаки применяются для предсказания будущих признаков восприятия, которые затем сравниваются с фактическими признаками восприятия из следующего кадра. Модуль *обнаружения ошибок* вычисляет ошибку предсказания, которая генерирует *сигнал сброса* (обозначен символом ★) при больших ошибках предсказания. Сигнал сброса обновляет *модель событий*, принимая входные данные от блоков *сенсорного входа* и *обработки восприятия*

представляет собой постоянный процесс восприятия, который сегментирует непрерывный поток мультимодальных сенсорных входных данных на связную последовательность дискретных событий. Этот набор может быть дополнительно сегментирован для формирования иерархии событий в различных временных масштабах. Важно подчеркнуть, что процесс сегментации событий не требует сознательного внимания. Вместо этого сегментация возникает как побочный эффект непрерывного процесса перцептивного предсказания.

По мере восприятия сенсорных входных данных перцептивный процессор получает, фильтрует и кодирует поступающие признаки в полезные представления более высокого уровня. В вычислительном отношении перцептивный процессор может быть представлен стеком глубокого обучения для обработки и извлечения признаков. Ключевым аспектом перцептивной обработки в EST является то, что закодированные признаки зависят от представлений из рабочей модели событий. Это условие делает представления признаков устойчивыми к небольшим изменениям от одного момента к другому. Рабочая модель события управляет перцептивной обработкой для извлечения соответствующих признаков наблюдаемого события. Рабочие модели событий очень специфичны и ограничены в возможностях; они обновляются на границах событий.

Блок обработки восприятия отправляет извлеченные признаки (обусловленные моделью рабочего события) в блок прогнозирования, чтобы предвидеть будущие признаки восприятия. Любое несоответствие между предсказанными и фактическими признаками для следующего момента времени постоянно отслеживается и называется *ошибкой предсказания*. Ошибка предсказания является мерой качества прогнозов и, следовательно, индикатором пригодности рабочей модели событий. Временное резкое увеличение ошибки предсказания является индикатором границы события, т. е. сигналом о том, что текущее событие могло измениться. Для продолжения прогнозирования необходима новая модель событий. Таким образом, ошибка предсказания работает как механизм стробирования для обновления рабочей модели представлением нового события. Это обновление рабочей модели события состоит из сенсорной информации и предварительных ожиданий из долговременной памяти о следующем событии. Долговременная память событий представлена *схемами событий*, которые кодируют более стабильное представление события по сравнению с рабочей моделью событий. Схемы событий хранят последовательную информацию с точки зрения отличительных физических характеристик объектов и действующих лиц, вероятных следующих событий и целей действующих лиц, причем все эти сведения извлекаются из ранее наблюдавшихся событий. Изменения в схемах событий происходят с меньшей скоростью обучения, чем в рабочей памяти событий.

Качество предсказания зависит от того, насколько точно рабочая модель события представляет восприятие. Как правило, рабочая модель хорошо настроена, а ошибка прогнозирования невелика. Однако время от времени текущие наблюдения могут становиться менее предсказуемыми, что приводит к увеличению ошибки предсказания и необходимости обновления модели

рабочего события. В структуре EST это обновление опосредуется через механизм стробирования как функцию сигнала ошибки. Когда сигнал ошибки увеличивается, рабочая модель события обновляется на основе сенсорного сигнала и информации из схемы событий до тех пор, пока ошибка не уменьшится. Таким образом, система в целом работает в основном в стабильном состоянии с низкими ошибками прогнозирования и переходными периодами высоких ошибок, сигнализирующих о границах событий.

12.3. ВАРИАНТ 1: ОДНОПРОХОДНАЯ СЕГМЕНТАЦИЯ ВО ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ ПРЕДСКАЗАНИЯ

Теория сегментации событий предлагает нам механизм сегментации событий во времени, т. е. разбиения видео на части без необходимости обучающих меток и в непрерывном онлайн-режиме. В этом разделе мы демонстрируем возможности EST, используя простую однопроходную реализацию EST, которая превосходит более сложные современные методы глубокого обучения с учителем.

В вычислительном отношении мы реализуем структуру EST, используя известные компоненты глубокого обучения. Блок перцептивного предсказания выполнен в виде модуля кодера на основе сверточных нейронных сетей (CNN) (LeCun et al., 1995). Ячейка *долгой краткосрочной памяти* (long short-term memory, LSTM) (Hochreiter, Schmidhuber, 1997) используется для агрегирования признаков во времени и прогнозирования будущих характеристик восприятия. Внутренняя структура ячеек LSTM предлагает идеальную альтернативу реализации для прогнозирования перцептивных признаков по двум причинам. Во-первых, рекуррентный характер LSTM позволяет нам интегрировать прошлые кадры, чтобы предсказать будущее. Во-вторых, скрытое состояние LSTM действует как внутренняя модель событий, которую можно использовать для привязки извлеченных признаков к стабильному представлению событий, построенному из предыдущих кадров. Механизм адаптивного обучения обеспечивает простой и практичный метод реализации механизма стробирования, используемого для обновления модели событий на основе ошибки перцептивного предсказания.

В работе (Aakur, Sarkar, 2019) сосредоточились на построении представлений для рабочей модели событий и не реализовывали модуль схем событий. Мы начнем с обсуждения кодирования кадров и извлечения признаков в разделе 12.3.1, после чего представим объяснение того, как мы используем рекуррентную ячейку для постоянного вычисления представления предыдущих кадров (раздел 12.3.2). Мы также кратко обсудим роль слоя реконструкции в преобразовании предсказанного представления в будущие перцептивные признаки в разделе 12.3.3. В разделах 12.3.5 и 12.3.6 мы представляем механизм вентилей и функции адаптивного обучения для обнаружения границ. В алгоритме 1 показан псевдокод структуры перцептивного предсказания.

Алгоритм 1. Модель временной сегментации событий. Ввод представляет собой необрезанное/потокоевое видео \mathbb{I} , которое представляет собой множество кадров $\{I_1, \dots, I_t, I_{t+1}, \dots, I_T\}$. На выходе получается множество границ событий $\{b_1, b_2, \dots, b_{T-1}\}$

Вход: Видеокадры $\{I_1, \dots, I_t, I_{t+1}, \dots, I_T\} \in \mathbb{R}^{T \times C \times W \times H}$

Выход: Значения границ события $\mathbb{B} = \{b_1, b_2, \dots, b_{T-1}\}$

1: **procedure** LSTM(h_{t-1}, I_t)

2: $i_t \leftarrow \sigma(W_i I_t + W_{hi} h_{t-1} + b_i)$

▷ Входной вентиль

3: $f_t \leftarrow \sigma(W_f I_t + W_{hf} h_{t-1} + b_f)$

▷ Вентиль забывания

4: $o_t \leftarrow \sigma(W_o I_t + W_{ho} h_{t-1} + b_o)$

▷ Выходной вентиль

5: $g_t \leftarrow \phi(W_g I_t + W_{hg} h_{t-1} + b_g)$

6: $m_t \leftarrow f_t \cdot m_{t-1} + i_t \cdot g_t$

7: $h_t \leftarrow o_t \cdot \phi(m_t)$

8: **return** h_t

9: **end procedure**

10: **procedure** GATE($E_p(t)$)

11: $P_q(t) = P_q(t-1) + \frac{1}{n}(E_p(t) - P_q(t-1))$

▷ Скользящее среднее

12: **if** $\frac{E_p(t)}{P_q(t-1)} > \omega_e$ **then**

13: $P_q(t-1) \leftarrow P_q(t)$

14: **return** *True*

15: **else**

16: $P_q(t-1) \leftarrow P_q(t)$

17: **return** *False*

18: **end if**

19: **end procedure**

20: **procedure** SEGMENT(I_t, I_{t+1}, h_{t-1})

▷ Главный слой сегментации

21: $I'_t \leftarrow \text{ENCODER}(I_t)$

▷ Базовый кодировщик CNN

22: $I'_{t+1} \leftarrow \text{ENCODER}(I_{t+1})$

▷ Базовый кодировщик CNN

23: $h_t \leftarrow \text{LSTM}(h_{t-1}, I'_t)$

24: $y_{t+1} \leftarrow \text{DECODER}(h_t)$

▷ Одиночный плотный слой

25: $E_p(t) \leftarrow \sum_{i=1}^n \|I'_{t+1} - y'_{t+1}\|_{\ell_i}^2$

26: $b_t \leftarrow \text{GATE}(E_p(t))$

27: **return** h_t, b_t

28: **end procedure**

29: $h_t \leftarrow 0$

30: **for** $\{I_t, I_{t+1}\} \in \{I_1, I_2\}, \{I_2, I_3\}, \dots, \{I_{T-1}, I_T\}$ **do**

31: $h_t, b_t \leftarrow \text{SEGMENT}(I_t, I_{t+1}, h_t)$

32: $\mathbb{B}.\text{append}(b_t)$

33: **end for**

В основе метода, показанного на рис. 12.3, лежит платформа прогнозирующей обработки, которая кодирует визуальный ввод I_t в абстракцию более высокого уровня I'_t , используя сеть кодировщика. Абстрактная функция используется в качестве априорной для прогнозирования функции I'_{t+1} в мо-

мент времени $t + 1$. Модуль предсказания будущего LSTM объединяет извлеченные признаки со стабильным представлением (скрытым состоянием) события на основе предыдущих кадров для предсказания будущего перцептивного представления. Сеть реконструкции или декодера преобразует предсказанное представление в фактические признаки (той же размерности, что и I_{t+1}), которые используются для обнаружения границ событий между последовательными действиями в потоковом входном видео.

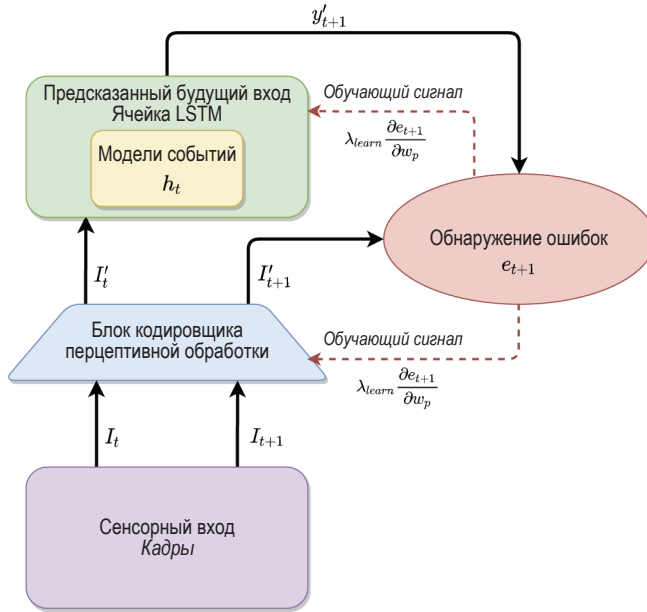


Рис. 12.3 ❖ Вариант 1: однопроходная сегментация во времени с использованием перцептивного предсказания. Архитектура модели представляет собой урезанную версию блоков, предложенных для EST. Она состоит из четырех основных компонентов: сети кодировщика, блока предсказания, сети декодирования, блока обнаружения ошибок и определения границ

12.3.1. Извлечение и кодирование признаков

Мы кодируем входной кадр на каждом временном шаге в абстрактные визуальные признаки более высокого уровня и используем эти признаки в качестве основы для перцептивной обработки вместо «сырого» ввода на уровне пикселей (для снижения сложности сети) или семантики более высокого уровня (которой нужны обучающие данные в виде меток). Оптимизированный кодировщик извлекает только признаки, относящиеся к решаемой задаче, в данном случае к прогнозированию. Процесс кодирования влечет за собой изучение функции $g(I(t), \omega_e)$, которая преобразует входной кадр I_t из пространства пикселей в пространство признаков более высокой размерности I'_t .

Функция преобразования $g(I(t), \omega_e)$ извлекает соответствующие пространственные признаки в сжатое векторное представление с использованием набора обучаемых параметров ω_e . На практике здесь можно использовать предварительно обученную базовую архитектуру для кодирования необработанных пикселей в полезные функции. В этом исследовании в качестве функции преобразования $g(\cdot)$ мы используем модель CNN VGG16 (Simonyan, Zisserman, 2014), предварительно обученную на наборе ImageNet (Russakovsky et al., 2015). Предварительно обученные веса используются для обеспечения хороших параметров инициализации, но дополнительно настраиваются с применением ошибки прогнозирования.

12.3.2. Рекуррентное прогнозирование для прогнозирования признаков

Итак, у нас есть перцептивные признаки в момент t (I'_t), и следующим шагом является предсказание перцептивных признаков в момент $t + 1$. Прогнозируемые признаки являются функцией извлеченных признаков I'_t и внутренней рабочей моделью текущего события. Внутренняя модель обрабатывает входной сенсорный сигнал в каждом кадре, подобно блоку обработки восприятия модели события в разделе 12.2. Формально этот процесс можно описать как генеративную модель $P(I'_{t+1} | \omega_p, I_t)$, где ω_p – множество скрытых параметров, характеризующих внутреннее состояние текущего наблюдаемого события.

Чтобы извлечь временные зависимости между кадрами *внутри* событий и кадрами *между* событиями, мы используем сеть LSTM (Hochreiter, Schmidhuber, 1997). Математическая форма модели LSTM представлена в алгоритме 1, где σ – нелинейная функция активации, точечный оператор (\cdot) обозначает умножение Адамара (поэлементное), ϕ – гиперболическая функция тангенса (\tanh), а W_x и b_x представляют обученные веса и наклоны для каждого из вентилях. В совокупности $\{W_{hi}, W_{hf}, W_{ho}, W_{hg}\}$ и их соответствующие смещения составляют обучаемые параметры ω_p .

Ячейка LSTM, формально определенная в алгоритме 1, состоит из трех основных вентилях: входного i_t , забывания f_t и выходного o_t . Вентили забывания и входа (в сочетании со слоем памяти g_t) работают вместе, чтобы обновлять внутреннюю модель событий в соответствии с их обучаемыми параметрами. Выходной вентиль обрабатывает пок кадровый входной сигнал восприятия во внутренней памяти текущего события. Стоит отметить, что можно использовать и другие рекуррентные модели, такие как *управляемый рекуррентный блок* (gated recurrent unit, GRU).

Состояние события h_t является представлением события, наблюдаемого в момент времени t , и, следовательно, более чувствительно к наблюдаемому входному сигналу $I'(t)$, чем слой событий, который более устойчив во всех событиях. Слой событий является стробируемым слоем, который получает входные данные от кодировщика и модели рекуррентных событий. Однако входные данные слоя событий модулируются самообучаемым стробирующим сигналом (раздел 12.3.5), что свидетельствует о качестве прогнозов, сде-

ланных рекуррентной моделью. Стробирование позволяет быстро обновлять веса, но также поддерживает согласованное состояние в рамках события.

12.3.3. Реконструкция признаков

Цель перцептивного процессора (или, скорее, сети реконструкции) состоит в том, чтобы реконструировать предсказанный признак y'_{t+1} по исходному предсказанию h_t , что максимизирует вероятность

$$p(y'_{t+1}|h_t) \propto p(h_t|y'_{t+1})p(y'_{t+1}), \quad (12.1)$$

где первый член – это вероятность, а второй – априорная модель признаков (feature prior model). Однако мы моделируем $\log p(y'_{t+1}|h_t)$ как логарифмически линейную модель $f(\cdot)$, зависящую от весов рекуррентной модели ω_p и наблюдаемого признака I'_t и характеризуемую уравнением

$$\log p(y'_{t+1}|h_t) = \sum_{n=1}^t f(\omega_p, I'_t) + \log Z(h_t), \quad (12.2)$$

где $Z(h_t)$ – нормировочная константа, не зависящая от весов ω_p и используемая для получения более конкретной оценки вероятности с учетом неопределенности. На практике ее игнорируют, поскольку предиктивное обучение обеспечивает необходимую регуляризацию. Модель реконструкции завершает генеративный процесс для прогнозирования функции в момент времени $t + 1$ и помогает замкнуть цикл самообучения для определения границ событий.

12.3.4. Функция потерь при самообучении

Одно из отличительных свойств признаков одного и того же события заключается в том, что они предсказуемы с учетом предыдущих входных данных восприятия и стабильной модели события. Мы используем это свойство для обнаружения границ событий. Следовательно, можем определить функцию ошибки прогнозирования, которая получает прогноз модели y'_t в качестве входных данных и фактические следующие сенсорные признаки I'_t в качестве целей. Результирующая ошибка, называемая *ошибкой перцептивного предсказания* (perceptual prediction error), рассчитывается, как показано в уравнении (12.3):

$$E_p(t) = \sum_{i=1}^n \|I'_t - y'_t\|_{\ell_2}^2. \quad (12.3)$$

Ошибка перцептивного предсказания обоснованно указывает на степень релевантности внутреннего состояния рекуррентной модели реальному наблюдаемому событию. Когда событие изменяется путем пересечения границы события, внутренняя модель становится непригодной для использования,

что приводит к неправильным прогнозам. Качество предсказания повышается после обновления внутренней модели более новым представлением, способным предсказывать следующие признаки. На рис. 12.4 представлен пример этого эффекта. Минимизация ошибки предсказания служит целевой функцией для обучения сети.

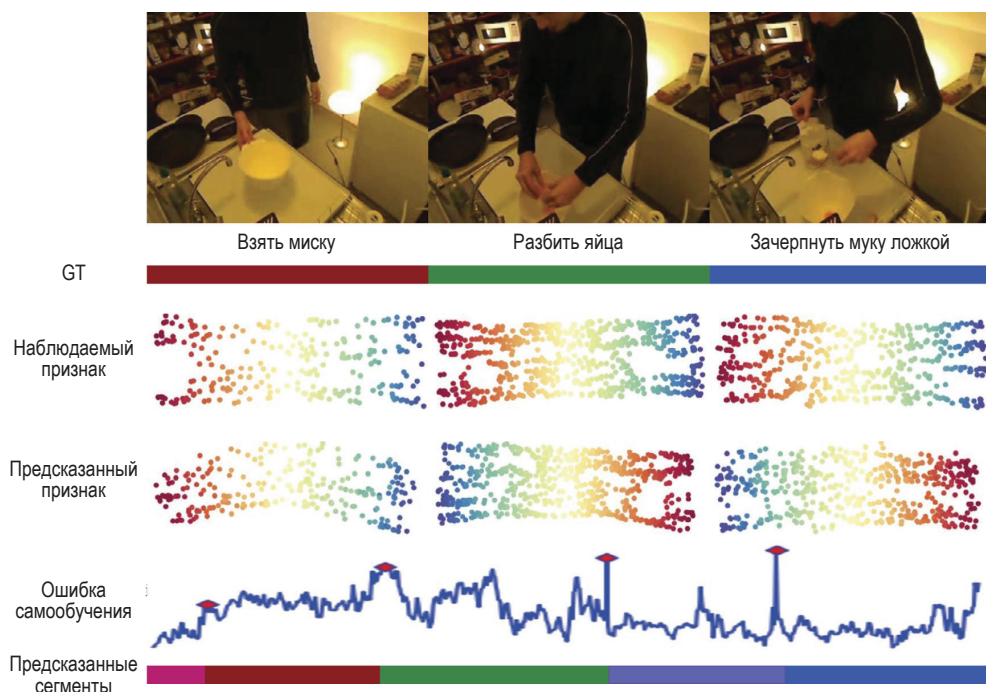


Рис. 12.4 ❖ Здесь показана визуализация процесса прогнозирующего обучения в теории сегментации событий. Текущие сенсорные входные данные абстрагируются в признаки меньшей изменчивости, на которых основано предсказание. Строится рабочая модель событий, которая используется для непрерывного предсказания признаков, наблюдаемых на следующем временном шаге. Предсказания постоянно сравниваются с наблюдаемыми признаками, и результирующая ошибка предсказания служит индикатором пригодности модели событий. Механизм стробирования, роль которого играет функция ошибки предсказания, модулирует процесс обучения и предоставляет сигналы для сегментации событий. Символ ♦ обозначает на нижнем графике предсказанные границы событий. Признаки были визуализированы с использованием T-SNE (van der Maaten, Hinton, 2008) для презентации. Воспроизведено с разрешения Аакура и Саркара (Aakur, Sarkar, 2019)

12.3.5. Механизм стробирования на основе ошибок

Согласно теории сегментации событий, перцептивные признаки могут стать крайне непредсказуемыми на границах событий, поскольку для обработки

нового события необходимо будет обновить рабочую модель. Например, на рис. 12.4 мы видим, что визуальное представление признаков, изученных сестрой кодировщика для действий «брать миску» и «разбивать яйца», ближе друг к другу, чем для признаков действий «брать миску» и «брать ложку муки». На границах событий рекуррентная внутренняя модель теряет способность объяснять расходящееся пространство признаков, вызывая временное увеличение ошибки перцептивного предсказания. Мы видим, что ошибка предсказания (второй график снизу) хорошо согласуется с сегментацией эталона (второй сверху) для видео «Выпекание блина». Как показано, частота ошибок выше на границах событий и ниже внутри события.

Чтобы смоделировать ошибку предсказания и найти границы событий, мы можем использовать фильтр нижних частот для нахождения скользящего среднего значения ошибки предсказаний, сделанных за последние n входных интервалов времени. Мы используем значение $n = 5$, исходя из среднего времени отклика человеческого восприятия (200 мс) (Thorpe et al., 1996). Находим скользящее среднее значение ошибки прогноза, называемое *качеством предсказания* и определяемое как

$$P_q(t) = P_q(t-1) + \frac{1}{n}(E_p(t) - P_q(t-1)), \quad (12.4)$$

где P_q – качество предсказания, а $E_p(t)$ – ошибка предсказания по уравнению (12.3). Сигнал стробирования модели $G(t)$ вырабатывается, когда текущая ошибка предсказания превышает средний показатель качества предсказания не менее чем на 50 %. Формально мы можем определить стробирующую функцию как

$$G(t) = \begin{cases} 1, & \frac{E_p(t)}{P_q(t-1)} > \psi_e, \\ 0 & \text{в ином случае} \end{cases}, \quad (12.5)$$

где $E_p(t)$ – ошибка предсказания в момент времени t , $G(t)$ – значение стробирующего сигнала в момент времени t , $P_q(t-1)$ – средняя метрика качества предсказания в момент времени t , а ψ_e – порог ошибки предсказания для обнаружения границ. Для оптимального прогнозирования ошибка перцептивного предсказания должна быть очень высокой в граничных кадрах события и очень низкой во всех кадрах внутри события. ψ_e – это гиперпараметр, который можно использовать для настройки временной шкалы, в которой мы будем обнаруживать границы событий.

12.3.6. Адаптивное обучение для повышения робастности

Изучение робастного представления событий лежит в основе подходов к сегментации событий с использованием предиктивного обучения. Представление события считается робастным, когда ошибка предсказания низка для

кадров *внутри* событий и высока для кадров *между* событиями. Если существует переобучение представления события на наблюдениях внутри события, то незначительные возмущения в необработанном пространстве пикселей могут увеличить ошибку прогнозирования. Это опровергло бы лежащее в основе предположение о том, что на изменение наблюдаемого события указывают временные ошибки. Кроме того, представление событий должно быть стабильным для событий с различной временной продолжительностью, чтобы избежать катастрофического забывания, т. е. ситуации, когда предсказания перестают отражать внутрисобытийные вариации в длинных последовательностях событий. Следовательно, необходимо гарантировать, что модель не будет переобучаться краткосрочным признакам восприятия, сохраняя при этом робастное представление события *целиком*.

Чтобы обеспечить определенную пластичность и избежать катастрофического забывания в сети, мы используем адаптивное обучение. Адаптивное обучение похоже на обучение с переменной скоростью, широко используемый метод обучения глубоких нейронных сетей. Однако вместо использования заранее определенных интервалов для изменения скорости обучения мы управляем скоростью, исходя из величины ошибки предсказания. Скорость обучения можно настраивать, чтобы управлять распространением ошибки обратно на обучаемые параметры.

Например, когда перцептивная скорость предсказания ниже средней скорости предсказания, прогнозная модель считается хорошим, стабильным представлением текущего события. Распространение ошибки предсказания при наличии хорошего представления события может привести к переобучению прогнозной модели для этого конкретного события и не способствует обобщению. Следовательно, более низкие скорости обучения используются для временных шагов, когда существует незначительная ошибка прогнозирования, а относительно более высокие – для более высоких ошибок прогнозирования.

Очевидно, что переменная скорость обучения позволяет модели намного быстрее адаптироваться к новым событиям (на границах событий, где вероятность ошибок выше) и научиться сохранять внутреннее представление для кадров внутри событий. Скорость обучения определяется как результат правила адаптивного обучения, описанного как функция ошибки перцептивного предсказания, определенной в разделе 12.3.4, и может быть записана в виде:

$$\lambda_{learn} = \begin{cases} \Delta_t^- \lambda_{init}, & E_p(t) > \varepsilon_e \\ \Delta_t^+ \lambda_{init}, & E_p(t) < \varepsilon_e \\ \lambda_{init} & \text{в ином случае} \end{cases}, \quad (12.6)$$

где Δ_t^- , Δ_t^+ и λ_{init} обозначают масштабирование скорости обучения в отрицательном направлении, положительном направлении и начальную скорость обучения соответственно, и $\varepsilon_e = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} E_p dt$. Скорость обучения регулируется исходя из качества предсказания, характеризуемого ошибкой пред-

сказания на временной последовательности между моментами времени t_1 и t_2 , обычно определяемой стробирующим сигналом.

12.3.7. Промежуточный итог

12.3.7.1. Наборы данных

Мы оцениваем и анализируем эффективность системы перцептивного предсказания на трех больших общедоступных наборах данных – «Действия за завтраком» (Kuehne et al., 2014), наборе данных INRIA (Alayrac et al., 2016) и наборе данных «50 салатов» (Stein, Маккенна, 2013). Каждый набор данных предлагает разные задачи, что позволяет нам оценить эффективность подхода в различных сложных условиях.

Набор данных о действиях за завтраком представляет собой большую коллекцию из 1712 видеороликов о 10 разновидностях деятельности за завтраком, выполненных 52 субъектами-актерами. Каждая деятельность состоит из нескольких подвидов деятельности, которые имеют визуальные и временные вариации в зависимости от предпочтений и стиля субъекта. Задачу сегментации во времени усложняют различия в качестве визуальных данных и такие помехи, как окклюзии и переменные ракурсы.

Набор обучающих видеороликов INRIA содержит 150 видеороликов о 5 различных видах деятельности, собранных с YouTube. Каждое из видео длится в среднем 2 минуты и содержит около 47 дополнительных подвидов деятельности. «Фоновый» класс обозначает последовательность, в которой не существует четкой деятельности, различимой визуально. Это создает серьезную проблему для методов, которые явно не обучены таким визуальным признакам.

Набор данных «50 салатов» представляет собой мультимодальный набор данных, собранный в области кулинарии. Набор данных содержит более четырех часов аннотированных данных для 25 человек, каждый из которых готовит по два смешанных салата. Он предоставляет данные в различных модальностях, таких как кадры RGB, карты глубины и данные акселерометров, прикрепленных к различным предметам, таким как ножи, ложки и бутылки (это лишь некоторые из них). Аннотации действий предоставлены на разных уровнях детализации – высоком, низком и оценочном. Мы используем оценочный уровень «eval» в соответствии с протоколами оценки в предыдущих работах (Lea et al., 2016, 2017).

12.3.7.2. Метрики оценки

Для анализа эффективности рассматриваемого подхода мы используем две широко используемые метрики оценки. Мы применяем тот же протокол оценки и код, что и в (Alayrac et al., 2016; Sener, Yao, 2018). По причине обучения без учителя при оценке точности используем венгерский алгоритм сравнения для получения взаимно однозначных сопоставлений между предсказанными сегментами и эталонами. Мы используем среднее значение по кадрам (mean

over frames, MoF) для оценки способности сети временно локализовать подкатегории деятельности. Оцениваем расхождение предсказанных сегментов с эталонной сегментацией с использованием индекса Жаккара (пересечение поверх объединения, или IoU). Мы также используем показатель F1 для оценки качества сегментации. Для оценки задачи распознавания в разделе 12.3.7.4.1 применяется критерий точности на уровне объектов для 48 классов, как показано в табл. 3 в статье (Kuehne et al., 2014) и сравнивается с (Kuehne et al., 2014; Aakur et al., 2019; de Souza et al., 2016; Huang et al., 2016).

12.3.7.3. Вариативные исследования

Мы оценивали различные варианты нашей структуры, чтобы сравнить эффективность каждого компонента. Мы варьировали историю предсказания n и порог ошибки предсказания Ψ . Увеличение окна кадра приводит к объединению кадров и меньших кластеров вблизи границ событий с предыдущим классом деятельности из-за временного увеличения ошибки. Это приводит к более высокому IoU и более низкому MoF. Низкий порог ошибки приводит к чрезмерной сегментации, поскольку механизм обнаружения границ становится чувствительным к небольшим изменениям. Количество предсказанных кластеров уменьшается по мере увеличения размера окна и порога. Мы также обучили четыре модели (рис. 12.5) с разными модулями предикторов. Обучили две рекуррентные нейронные сети (RNN) в качестве предикторов с адаптивным обучением (AL) и без него, описанные в разделе 12.3.6, обозначенные как RNN + No AL и RNN + AL соответственно. Мы также обучили LSTM без адаптивного обучения (LSTM + No AL) для сравнения с нашей

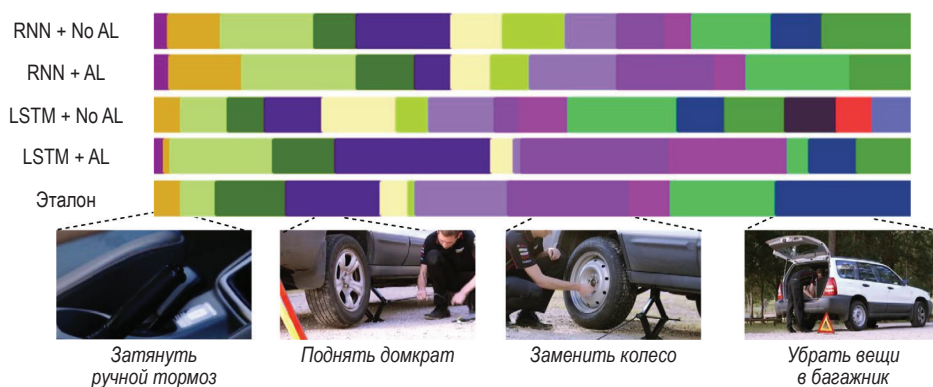


Рис. 12.5 ❖ Вариативные исследования: сравнение вариантов архитектуры модели с использованием RNN и LSTM с адаптивным обучением и без него в наборе учебных видео INRIA и видео с эталонным образцом Change Tite (замена покрышки). Можно видеть, что сложные визуальные сцены с действиями меньшей продолжительности создают серьезную проблему для модели и вызывают фрагментацию и чрезмерную сегментацию. Однако использование адаптивного обучения помогает в некоторой степени смягчить этот эффект. Примечание: шкалы сегментации по времени для лучшей наглядности показаны без фоновых классов

основной моделью (LSTM + AL). Мы используем RNN в качестве возможной альтернативы из-за необходимости краткосрочных предсказаний будущего (на 1 кадр вперед).

12.3.7.4. Количественная оценка

Набор данных «Действия во время завтрака». Мы оцениваем точность нашей полной модели LSTM + AL в наборе данных о действиях во время завтрака и сравниваем с подходами, использующими полное обучение с учителем, частичное обучение с учителем и обучение без учителя. Мы показываем эффективность подхода SVM (Kuehne et al., 2014), чтобы подчеркнуть важность кратковременного моделирования. Как показано в табл. 12.1, метод перцептивного предсказания превзошел все другие методы с обучением без учителя и с частичным привлечением учителя, а также некоторые подходы традиционного обучения с учителем.

Таблица 12.1. Результаты сегментации набора данных «Действия во время завтрака». MoF обозначает метрику «Среднее значение по кадрам», а IoU – метрику «Пересечение поверх объединения»

Разметка	Метод	MoF, %	IoU
Полная	SVM (Kuehne et al., 2014)	15,8	–
	HTK(64) (Kuehne et al., 2016)	56,3	–
	ED-TCN (Lea et al., 2017)	43,3	42,0
	TCFPN (Ding, Xu, 2018)	52,0	54,9
	GRU (Richard et al., 2017)	60,6	–
Слабая	OCDC (Bojanowski et al., 2014)	8,9	23,4
	ECTC (Huang et al., 2016)	27,7	–
	Fine2Coarse (Richard, Gall, 2016)	33,3	47,3
	TCFPN + ISBA (Ding, Xu, 2018)	38,4	40,6
Нет	KNN + GMM (Sener, Yao, 2018)	34,6	47,1
	Наш метод (LSTM + AL)	42,9	46,9

Следует отметить, что другой метод с обучением без учителя (Sener, Yao, 2018) для достижения заявленной точности требует наличия достаточно большого количества кластеров (из эталона). Напротив, наш подход не требует таких знаний и выполняется в потоковом режиме. Кроме того, методы с частичным обучением (Huang et al., 2016; Richard, Gall, 2016; Ding, Xu, 2018) требуют в качестве входных данных как большого количества действий, так и упорядоченного списка подвидов деятельности. ECTC (Huang et al., 2016) основан на дискриминативной кластеризации, в то время как OCDC (Bojanowski et al., 2014) и Fine2Coarse (Richard, Gall, 2016) основаны на RNN.

Набор данных «50 салатов». Мы также оценили наш метод на наборе данных «50 салатов», используя в качестве входных данных только визуальные признаки. Для объективного сравнения представлен показатель среднего значения по кадрам (MoF). Как видно из табл. 12.2, наш предсказательный метод значительно превосходит другой метод с обучением без учителя,

улучшая показатель MoF на 6,6 %. Мы также приводим показатели методов классификации на основе кадров VGG и IDT (Lea et al., 2016), чтобы продемонстрировать эффективность кратковременного моделирования.

Таблица 12.2. Результаты сегментации набора данных «50 салатов» с уровнем детализации «Eval». **Результаты модели были преднамеренно представлены без временных ограничений для вариативного тестирования

Разметка	Метод	MoF, %
Полная	VGG** (Lea et al., 2016)	7,6
	IDT** (Lea et al., 2016)	54,3
	S-CNN + LSTM (Lea et al., 2016)	66,6
	TDRN (Lei, Todorovic, 2018)	68,1
	ST-CNN + Seg (Lea et al., 2016)	72,0
	TCN (Lea et al., 2017)	73,4
	LSTM + KNN (Bhatnagar et al., 2017)	54,0
Нет	Наш метод (LSTM + AL)	60,6

Следует отметить, что для методов обучения с учителем требуется значительно больше обучающих данных – как в виде меток, так и в виде обучающих эпох. Кроме того, метод TCN (Lea et al., 2017) для достижения точности 73,4 % использует данные акселерометра.

Набор учебных видео INRIA. Наконец, мы протестировали наш метод на наборе учебных видео INRIA, который создал серьезную проблему в виде большого количества фоновых (шумовых) данных. Была получена оценка F1 для справедливого сравнения с другими современными методами. Как видно из табл. 12.3, предсказательная модель превосходит другой метод с обучением без учителя (Sener, Yao, 2018) на 7,5 %, с частичным обучением (Bojanowski et al., 2014) на 7,9 % и конкурентоспособна в сравнении с методами, основанными на традиционном обучении с учителем (Malmaud et al., 2015; Alayrac et al., 2016; Sener, Yao, 2018).

Мы также оценили качество моделей с адаптивным обучением и без него. Как видно из табл. 12.3, эффективность LSTM в извлечении долгосрочных временных зависимостей значительна, в первую очередь из-за большой продолжительности действий в наборе данных. Кроме того, адаптивное обучение значительно улучшило структуру сегментации, повысив точность на 9 % и 11 % для модели на основе RNN и модели на основе LSTM соответственно, что указывает на уменьшение переобучения модели визуальными данными.

12.3.7.4.1. Улучшенные функции распознавания действий

Чтобы оценить способность сети изучать признаки с высокой степенью различения для последующего распознавания объектов, мы оценили эффективность предсказательного подхода в задаче распознавания. Мы предварительно обучаем модель для сегментации во времени, применяя набор данных «Действия во время завтрака», и используем скрытый слой LSTM в качестве источника входных данных для полностью подключенного слоя, минимизируя потери кросс-энтропии при обучении. Мы также обучили другую сеть

с такой же структурой – VGG16 LSTM – без предварительной подготовки, чтобы показать полезность изученных признаков при самообучении.

Таблица 12.3. Результаты сегментации набора обучающих видео INRIA.

Для справедливого сравнения использована оценка F1

Разметка	Метод	F1, %
Полная	HMM + Текст (Malmaud et al., 2015)	22,9
	Дискриминативная кластеризация (Alayrac et al., 2016)	41,4
	KNN + GMM (Sener, Yao, 2018) + GT	69,2
Слабая	OCDC + Текстовые признаки (Bojanowski et al., 2014)	28,9
	OCDC (Bojanowski et al., 2014)	31,8
Нет	KNN + GMM (Sener, Yao, 2018)	32,2
	Наш метод (RNN + No AL)	25,9
	Наш метод (RNN + AL)	29,4
	Наш метод (LSTM + No AL)	36,4
	Наш метод (LSTM + AL)	39,7

Как видно из табл. 12.4, использование самообучения для предварительного обучения сети перед задачей распознавания повышает точность распознавания сети и дает качество работы, сравнимое с другими современными методами. Предложенный нами подход повышает точность распознавания на 4,3 % с сетью, не подвергавшейся предварительному обучению предсказательной задаче.

Таблица 12.4. Результаты распознавания действий в наборе данных «Действия за завтраком». HCF и AL обозначают созданные вручную признаки и адаптивное обучение соответственно

Метод	Точность, %
HCF + HMM (Kuehne et al., 2014)	14,90
HCF + CFG + HMM (Kuehne et al., 2014)	31,8
RNN + ECTC (Huang et al., 2016)	35,6
RNN + ECTC (Cosine) (Huang et al., 2016)	36,7
HCF + Теория паттернов (de Souza et al., 2016)	38,6
HCF + Теория паттернов + ConceptNet (Aakur et al., 2019)	42,9
VGG16 + LSTM	33,54
VGG16 + LSTM + Предсказательные признаки (AL)	37,87

12.3.7.5. Качественная оценка

Благодаря прогнозной самообучаемой архитектуре мы можем изучить последовательность визуальных признаков в потоковом видео. Качество сегментации модели на наборе данных Breakfast Actions графически представлено на рис. 12.6. Можно видеть, что предсказательная архитектура демонстрирует хорошее совпадение с реальными отрезками времени и не страдает чрезмерной сегментацией, особенно когда сегменты длинные. Длинные после-

довательности действий позволяют модели учиться на основе наблюдения, предоставляя больше выборки «внутри события». Кроме того, методы с частичным обучением, такие как ODCD (Bojanowski et al., 2014) и ECTC (Huang et al., 2016), страдают чрезмерной сегментацией и фрагментацией внутри класса. Вероятно, это можно объяснить тем фактом, что они склонны навязывать семантику в виде упорядочения действий в видео независимо от изменений визуальных признаков. Методы на основе традиционного обучения с учителем, такие как НТК (Kuehne et al., 2016), работают лучше, особенно благодаря их способности назначать семантику визуальным признакам. Однако на них также влияют несбалансированные данные и смещение наборов данных, как видно на рис. 12.6, где фоновый класс был разбит на другие классы.

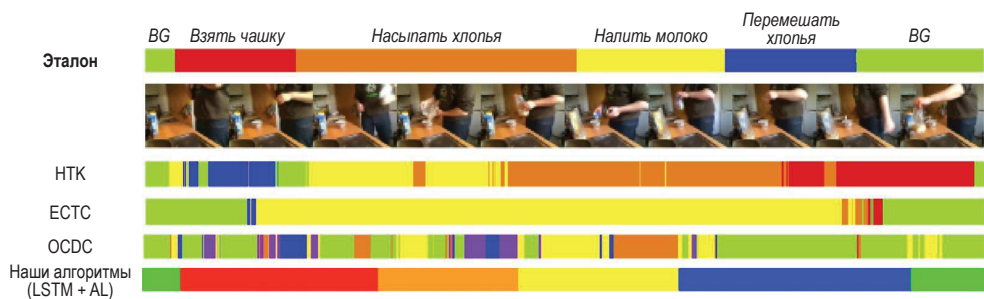


Рис. 12.6 ❖ Графическое представление точности сегментации основной модели в наборе данных «Действия за завтраком» на видео с эталонной информацией «Приготовление хлопьев». Предсказательный подход не склонен к излишней сегментации и обеспечивает логичную последовательную сегментацию. Однако видно, что для определения границ визуально похожих действий ему требуется больше времени

Мы также провели качественную оценку влияния адаптивного обучения и долговременной временной памяти в тесте, который был представлен на рис. 12.5, а соответствующие альтернативные методы описаны в разделе 12.2. Можно видеть, что использование адаптивного обучения предотвращает переобучение модели на внутрисобытийных кадрах любого отдельного класса и помогает обобщать другие классы независимо от объема обучающих данных. Нельзя сказать, что это решает проблему несбалансированных данных, но адаптивное обучение в некоторой степени приносит пользу.

12.4. ВАРИАНТ 2: СЕГМЕНТАЦИЯ С ИСПОЛЬЗОВАНИЕМ МОДЕЛЕЙ СОБЫТИЙ НА ОСНОВЕ ВНИМАНИЯ

Простая архитектура перцептивного предсказания, представленная в разделе 12.3, использует для прогнозирования *глобальное* представление входных видеок кадров. Она не сосредоточивается на конкретных пространственных областях как для восприятия, так и для прогнозирования и может вообще не

замечать детализированные представления событий как таковые. Если бы модели событий могли влиять на перцептивную обработку текущего сенсорного ввода, это существенно дополнило бы возможности EST (рис. 12.2). В нашей предыдущей модели эта возможность была косвенно реализована через структуру памяти в LSTM. В этом разделе мы покажем, что инфраструктура модели может быть дополнена идеей *пространственного внимания* (spatial attention), чтобы помочь ей сосредоточиться на ограниченных пространственных областях для более детализованного подхода к сегментации, что позволит обрабатывать очень длинные видеопоследовательности *без потребности в значительном обучении*. Полная архитектура модели детально изображена на рис. 12.7 и формально выражена в алгоритме 2.

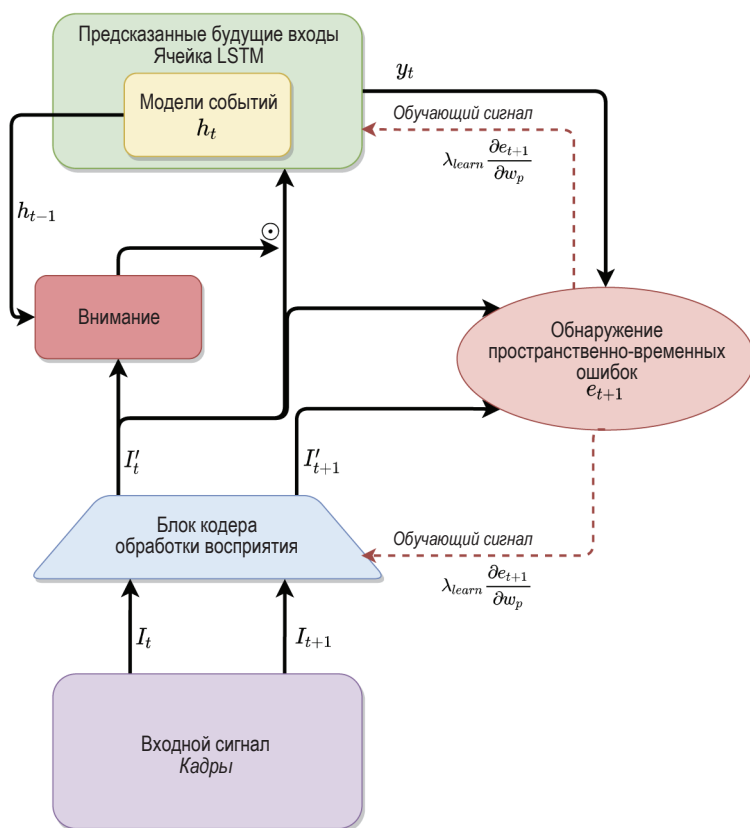


Рис. 12.7 ❖ Вариант 2: сегментация действий во времени с использованием моделей событий, основанных на внимании. Архитектура самообучающегося алгоритма перцептивного предсказания дополнена механизмом внимания, который предлагает более сильный способ воздействия модели события на текущую сенсорную информацию. Входные кадры на каждый момент времени кодируются в признаки высокого уровня с использованием стека глубокого обучения, за которым следует наложение механизма внимания на основе входных данных из предыдущих моментов времени, которые вводятся в LSTM. Потери при обучении вычисляются на основе разницы предсказанных и вычисленных признаков текущего и следующего кадров

Механизм пространственного внимания помогает сети изучить функцию внимания между внутренней моделью текущего события в рекуррентной памяти и входными данными на каждом такте времени. Полученную карту внимания можно использовать для пространственной локализации события в каждом обрабатываемом кадре.

Алгоритм 2. Модель сегментации событий во времени с пространственной локализацией на основе внимания. Ввод представляет собой необрезанное/потокковое видео \mathbb{I} , которое представляет собой множество кадров $\{I_1, \dots, I_t, I_{t+1}, \dots, I_T\}$. На выходе получается множество границ событий $\{b_1, b_2, \dots, b_{T-1}\}$

Вход: Видеокадры $\{I_1, \dots, I_t, I_{t+1}, \dots, I_T\} \in \mathbb{R}^{T \times C \times W \times H}$

Выход: Значения границ события $\mathbb{B} = \{b_1, b_2, \dots, b_{T-1}\}$

```

1: procedure ATTENTION( $I'_t, h_{t-1}$ )                                     ▷ Модуль внимания
2:  $a_t \leftarrow \text{linear}(\tanh(\text{linear}(h_{t-1}) + \text{linear}(I'_t)))$ 
3:  $A_t \leftarrow \text{softmax}(a_t)$ 
4:  $I''_t \leftarrow A_t \odot I'_t$ 
5: return  $I''_t$ 
6: end procedure

7: procedure SEGMENT( $I_t, I_{t+1}, h_{t-1}, y_{t-1}$ )                             ▷ Главный слой сегментации
8:  $I'_t \leftarrow \text{ENCODER}(I_t)$                                            ▷ Базовый кодировщик CNN
9:  $I'_{t+1} \leftarrow \text{ENCODER}(I_{t+1})$                                      ▷ Базовый кодировщик CNN
10:  $I''_t \leftarrow \text{ATTENTION}(I'_t, h_{t-1})$ 
11:  $h_t \leftarrow \text{LSTM}(h_{t-1}, \text{linear}(\text{concat}(I''_t, y_{t-1})))$ 
12:  $y_t \leftarrow \text{DECODER}(h_t)$                                            ▷ Одиночный плотный слой
13:  $e_t \leftarrow \|(I'_{t+1} - y_t)^{\odot 2} \odot (I'_{t+1} - I'_t)^{\odot 2}\|^2$       ▷ Взвешенная потеря
14:  $b_t \leftarrow \text{GATE}(e_t)$ 
15: return  $h_t, b_t, y_t$ 
16: end procedure

17:  $h_t \leftarrow 0$ 
18:  $y_{t-1} \leftarrow 0$ 
19: for  $\{I_t, I_{t+1}\} \in \{I_1, I_2\}, \{I_2, I_3\}, \dots, \{I_{T-1}, I_T\}$  do
20:  $h_t, b_t, y_t \leftarrow \text{SEGMENT}(I_t, I_{t+1}, h_t, y_{t-1})$ 
21:  $\mathbb{B}.\text{append}(b_t)$ 
22: end for

```

12.4.1. Извлечение признаков

Процесс извлечения признаков в этой архитектуре отличается от рассмотренной ранее. Чтобы работал механизм внимания, нам нужно обрабатывать пространственное расположение векторов признаков. В этой архитектуре кодировщик выводит сетку векторов признаков с тем же пространственным разрешением, что и результирующая карта внимания. Другими словами, механизм внимания диктует значение внимания, присваиваемое каждому из закодированных векторов признаков. В кодировщике мы используем только

операции свертки, поскольку ядра (веса) поддерживают пространственную конфигурацию карт признаков.

Необработанные входные изображения преобразуются из пространства пикселей в пространство признаков более высокого уровня с использованием модели кодировщика (CNN). Это закодированное представление признаков позволяет сети извлекать признаки, более важные для изучаемой задачи. Сеть кодировщика преобразует входное изображение с размерами $W \times H \times D$ в выходные признаки с размерами $N \times N \times M$, где $N \times N$ – пространственные размеры, а M – длина вектора признаков.

12.4.2. Модуль внимания

Модули внимания успешно применяются в задачах, где есть обучение с учителем, таких как создание подписей к изображениям (Xu et al., 2015), а также для различных задач обработки естественного языка, таких как перевод и языковое моделирование (Vaswani et al., 2017; Bahdanau et al., 2014; Luong et al., 2015; Devlin et al., 2018; Ян и др., 2019). В авторегрессионных языковых моделях внимание используется для предоставления декодирующей рекуррентной ячейке различных временных или пространственных сегментов ввода на каждом временном шаге. Мы используем внимание в несколько иной форме, когда LSTM декодируется только один раз (для каждого входного кадра) для предсказания будущих признаков и использует для этого взвешенный по вниманию ввод. В отличие от (Xu et al., 2015; Vaswani et al., 2017; Bahdanau et al., 2014; Luong et al., 2015; Devlin et al., 2018; Yang et al., 2019), веса внимания обучаются с использованием функций потерь обучения без учителя.

В рассматриваемой архитектуре мы используем внимание Бахданау (Bahdanau et al., 2014, 2014) для пространственной локализации события в каждом обработанном кадре. Блок внимания получает в качестве входных данных закодированные признаки и выводит набор весов внимания (A_t) с размерами $N \times N \times 1$. Скрытые векторы признаков (h_{t-1}) из слоя предсказания предыдущего временного шага используются для вычисления выходных весов внимания (изображенных на рис. 12.7) как

$$A_t = \gamma(FC(\tanh(FC(h_{t-1}) + FC(I_t''))), \quad (12.7)$$

где FC представляет собой один полносвязный слой нейронной сети, а γ – функцию softmax. Затем веса (A_t) умножаются на закодированные входные векторы признаков (I_t') для создания маскированных векторов признаков (I_t''). Некоторая визуализация результирующих масок пространственного внимания показана на рис. 12.8. Маска внимания извлекается из A_t , линейно масштабируется, а затем накладывается на необработанное входное изображение (I_t).

12.4.3. Функция потерь, взвешенная по движению

Потери предсказания, которые мы обсуждали в разделе 12.3.4, применяют функцию потерь L2 к предсказанию всего кадра. В этом разделе мы вводим

взвешенную по движению функцию потерь (motion weighted loss function), которая извлекает признаки, связанные с движением, из векторов признаков. Потери, взвешенные по движению, рассматриваются в пространстве признаков закодированных кадров, а не в «сыром» пространстве пикселей, и вычисляются с использованием непрерывной маски, применяемой к потерям предсказания. Эта модификация обеспечивает увеличение потерь предсказания для движущихся объектов при одновременном снижении потерь статических/фоновых объектов. На рис. 12.8 показано сравнение предсказания и взвешенных потерь движения для событий, когда птица входит в гнездо и выходит из него. Взвешенные потери при движении формально определяются как

$$e_t \leftarrow \|(I'_{t+1} - y_t)^{\odot 2} \odot (I'_{t+1} - I_t)^{\odot 2}\|^2, \quad (12.8)$$

где символ \odot обозначает операцию Адамара (поэлементно). Обратите внимание, что взвешенный по движению вектор (второй член) вычисляется *на уровне признаков*, а не на уровне пикселей, и, следовательно, более устойчив к незначительным изменениям из-за шума сенсора.

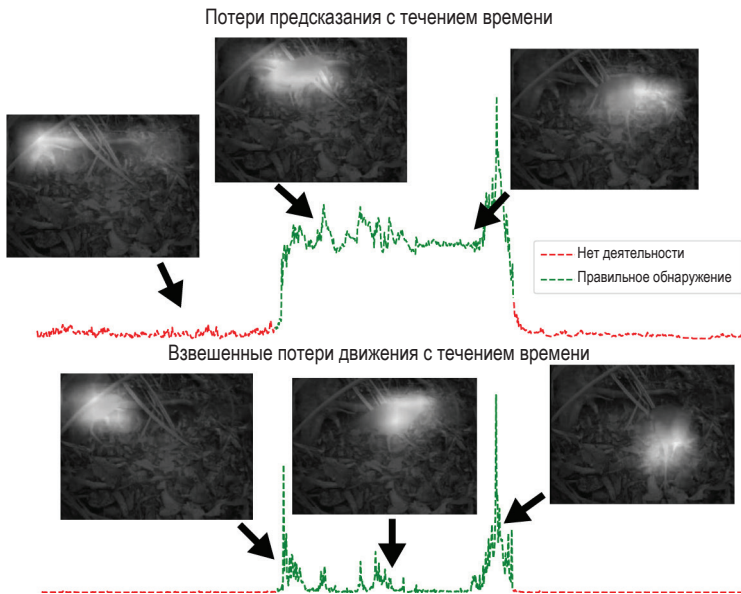


Рис. 12.8 ❖ Графики предсказания и потерь, взвешенных по движению, до, во время и после деятельности. Вверху: потери при предсказании признаков по кадрам. Внизу: потери при предсказании признаков, взвешенных по движению, по кадрам. Ошибки для некоторых выбранных кадров показаны на обоих графиках с наложением соответствующей карты внимания

12.4.4. Результаты

В этом разделе мы рассмотрим результаты обработки видео за несколько дней, с целью отметить пространственное положение и временные границы

событий. В отличие от других наборов данных, события в столь продолжительных видеороликах происходят только в нескольких кадрах, в то время как остальная часть видео не содержит значительных событий или движения (фоновых действий). Мы модифицируем фреймворк перцептивного предсказания, включив в него механизм внимания для пространственной локализации, а также маскируем функцию потери предсказания для извлечения признаков, связанных с движением. Начнем с описания расширенного набора видеоданных о дикой природе, используемого для тестирования модифицированного метода, после чего дадим пояснения по поводу показателей, используемых для количественной оценки качества модели. Мы обсудим оцененные варианты модели и в заключение представим количественные и качественные результаты.

12.4.4.1. Набор данных

Мы анализируем качество нашей модели на наборе данных мониторинга дикой природы. Набор данных состоит из 10 дней (254 часа) непрерывного наблюдения за гнездом кагу, нелетающей птицы из Новой Каледонии. Метки видео обозначают четыре уникальных вида деятельности птиц: {кормление птенца, инкубация/высиживание, строительство гнезда, сидение в гнезде, достраивание гнезда}. Вместе с метками для каждого экземпляра упомянутых видов деятельности предоставляется время начала и окончания. Мы изменили метки, добавив в них события «вход» и «выход», представляющие переходные события от пустого гнезда к инкубации и наоборот. Наш вариант может включать строительство гнезда (начальный садок и достраивание), кормление птенца, вход и выход. Другие события, основанные на климате, времени суток, условиях освещения, игнорируются нашей сетью сегментации. На рис. 12.9 показана выборка изображений из набора данных.



Рис. 12.9 ❖ Образцы изображений из набора данных мониторинга дикой природы (видеоролик про жизнь птиц кагу)

12.4.4.2. Критерии оценки

Количественные результаты представлены в виде графиков рабочих характеристик приемника (receiver operating characteristic, ROC) для сегментации событий как на уровне кадра (рис. 12.10), так и на уровне деятельности (рис. 12.11 и 12.12). Размер окна кадра (ϕ) определяется как максимальный размер окна относительно слияния событий; высокое значение ϕ может при-

вести к слиянию отдельных обнаруженных событий, что снижает общий показатель качества.

12.4.4.2.1. Уровень кадра

Значение полноты отклика в ROC на уровне кадра рассчитывается как отношение истинно положительных кадров (наличие события) к количеству положительных кадров в наборе данных, а доля ложных срабатываний выражается как отношение ложноположительных кадров к общему количеству отрицательных кадров (событие отсутствует) в наборе данных. Для построения одной линии ROC мы меняем пороговое значение (ψ), в то время как изменение размера окна кадра (ϕ) при неизменном значении порога приводит к построению другой линии ROC.

12.4.4.2.2 Уровень деятельности

Для однозначного сравнения событий, помеченных эталонной разметкой, и обнаруженных событий используется алгоритм венгерского сравнения (назначение Мункреса). Полнота отклика определяется как отношение количества правильно обнаруженных событий к общему количеству эталонных событий. Для построения графика ROC на уровне деятельности значения полноты отклика наносятся на график в зависимости от частоты ложноположительных результатов в минуту, определяемой как отношение общего количества обнаруженных ложноположительных событий к общей продолжительности видео из набора данных в минутах. Метрика оценки частоты ложноположительных результатов в минуту также используется в задаче ActEV TRECVID (ActEV, 2019) для количественной оценки систем обнаружения деятельности. Для построения одной линии ROC мы меняем пороговое значение (ψ), в то время как изменение размера окна кадра (ϕ) при неизменном значении порога приводит к построению другой линии ROC.

12.4.4.3. Вариативные исследования

Для количественной оценки влияния отдельных компонентов на общие показатели модели мы исследовали разные варианты нашей архитектуры. В наших экспериментах мы протестировали базовую модель, которая обучает систему перцептивного предсказания, включая блок внимания, с использованием функции потери предсказания для обратного распространения сигнала ошибки. Мы обозначили базовую модель как LSTM + ATTN. Мы также экспериментировали с влиянием удаления модуля внимания из архитектуры модели на общее качество сегментации; результаты этого варианта представлены под названием LSTM. Дальнейшее тестирование включает использование взвешенных по движению потерь для обратного распространения сигнала ошибки. Мы обозначили взвешенную по движению модель как LSTM + ATTN + MW. Каждая из моделей была тщательно протестирована; результаты приведены в разделах 12.4.4.4 и 12.4.4.5, а также изображены на рис. 12.10–12.14.

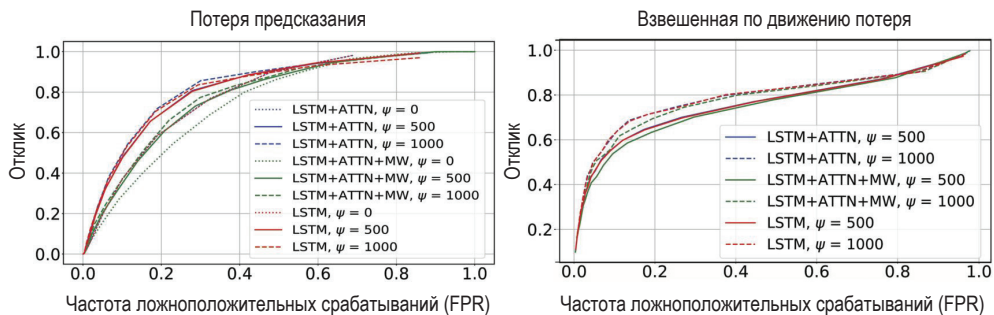


Рис. 12.10 ❖ ROC сегментации событий на уровне кадра, когда деятельности обнаруживаются на основе простой пороговой обработки сигналов предсказания и потерь, взвешенных по движению. Графики показаны для различных вариантов тестирования

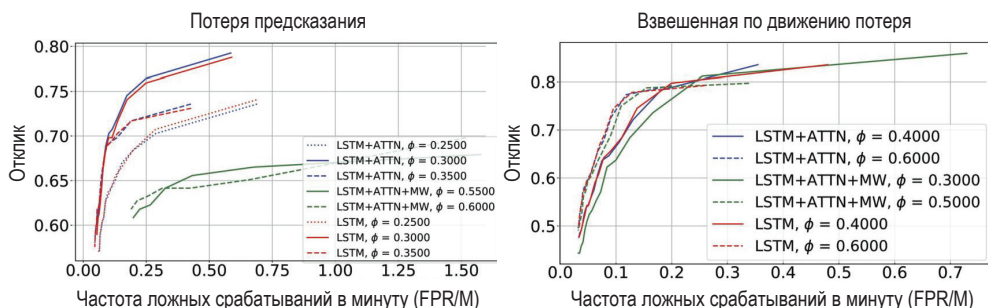


Рис. 12.11 ❖ ROC сегментации событий на уровне деятельности, когда деятельность обнаруживается на основе простой пороговой обработки сигналов предсказания и потерь, взвешенных по движению. Графики показаны для различных вариантов тестирования

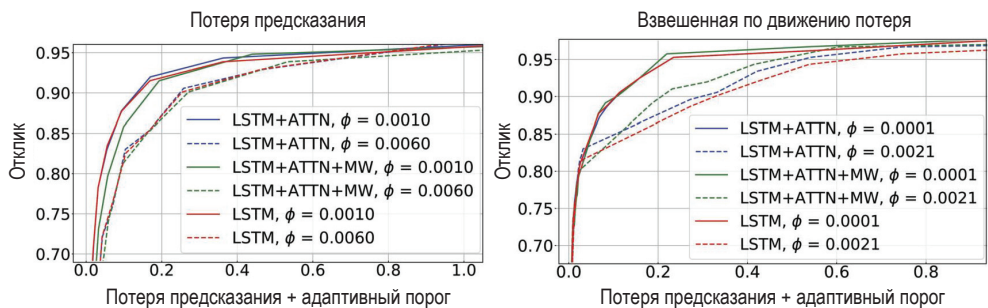


Рис. 12.12 ❖ ROC сегментации событий на уровне деятельности, когда деятельность обнаруживается на основе адаптивной пороговой обработки сигналов предсказания и потерь, взвешенных по движению. Графики показаны для различных вариантов тестирования

12.4.4.4. Количественная оценка

Мы обучили три разные модели, LSTM, LSTM + ATTN и LSTM + ATTN + MW, для сегментации событий на уровне кадров и на уровне деятельности. Для экспериментов на уровне кадров и уровне деятельности к предсказательным и взвешенным по движению сигналам потерь были применены простые и адаптивные функции стробирования (раздел 12.4.3). ROC-кривые для каждой модели, изображенные на рис. 12.10, 12.11 и 12.12, были получены путем изменения таких параметров, как пороговое значение ψ и размер окна кадра ϕ .

Следует отметить, что пороговое значение сигнала потерь не обязательно означает, что модель была обучена минимизировать этот конкретный сигнал. Другими словами, функции потерь, используемые для обратного распространения ошибки на обучаемые параметры моделей, указаны в имени модели (раздел 12.4.4.3); однако нами были проведены эксперименты по установлению порога для различных типов сигналов потерь, независимо от функции потерь обратного распространения, используемой для обучения.

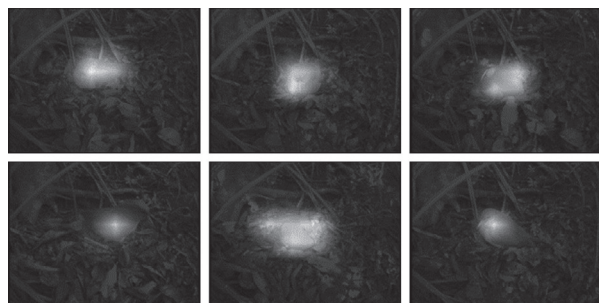
Модель с наилучшим качеством сегментации на уровне кадров (LSTM + ATTN, $\psi = 1000$) способна достичь значения полноты отклика {40 %, 60 %, 80 %} кадров при частоте ложных срабатываний {5 %, 10 %, 20 %} кадров соответственно. Сегментация на уровне деятельности может обеспечить полноту отклика {80 %, 90 %, 95 %} с частотой ложных обнаружений деятельности {0,02, 0,1, 0,2} в минуту соответственно для модели (LSTM + ATTN, $\phi = 0,0021$), как показано на рис. 12.12. Частоту ложноположительного обнаружения деятельности, равную 0,02 в минуту, также можно интерпретировать как ложное обнаружение одной деятельности каждые 50 минут обучения (с полнотой отклика 80 % для эталонной деятельности).

Сравнение результатов, изображенных на рис. 12.11 и 12.12, показывает значительное улучшение качества модели при использовании адаптивного порога для формирования стробирующего сигнала потери. Эффективность адаптивного порога очевидна при применении к сегментации событий на уровне активности. Результаты также показали, что модель может эффективно генерировать карты внимания (раздел 12.4.4.5) без ухудшения качества сегментации.

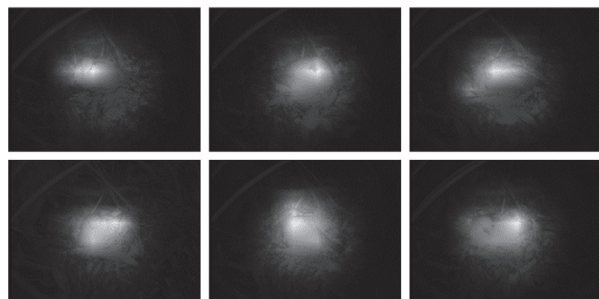
12.4.4.5. Качественная оценка

Примеры качественной оценки механизма внимания представлены на рис. 12.13 и 12.14. Маска внимания, извлеченная из модели, обучена без учителя отслеживать событие во всех обрабатываемых кадрах. Наши результаты показывают, что события отслеживаются и локализуются при различном освещении (тени, день/ночь) и в условиях окклюзии. Механизм внимания также научился бесконечно долго фокусироваться на птице независимо от ее состояния движения (подвижна/неподвижна). Это говорит о том, что модель приобрела высокоуровневое понимание событий в сцене и изучила базовую структуру птицы путем обучения без учителя. Дополнительные результаты¹

¹ Доступны по адресу <https://ramyamounir.github.io/projects/EventSegmentation>.

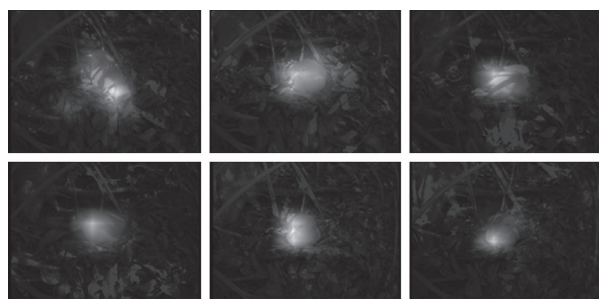


(a) Кадры днем, когда птица неподвижна

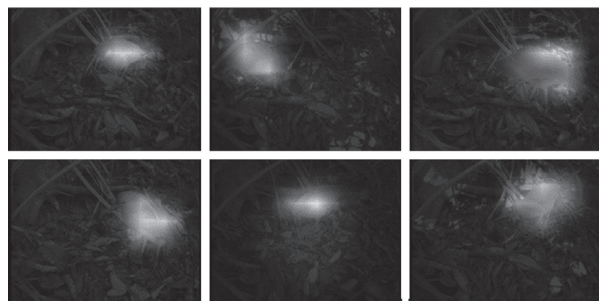


(b) Кадры ночью, когда птица неподвижна

Рис. 12.13 ❖ Примеры весов внимания Бахданау, визуализированные на входных изображениях



(a) Кадры в дневное время с движущимися тенями



(b) Кадры в дневное время во время движения птицы

Рис. 12.14 ❖ Примеры весов внимания Бахданау, визуализированные на входных изображениях

относятся к взвешенным по вниманию кадрам замедленной съемки во время изменения освещения и движущихся теней. На рис. 12.8 изображены сигнал предсказательной потери, сигнал потери, взвешенной по движению, и маска внимания во время событий входа в гнездо и выхода из него.

12.5. ВАРИАНТ 3: ПРОСТРАНСТВЕННО-ВРЕМЕННАЯ ЛОКАЛИЗАЦИЯ С ИСПОЛЬЗОВАНИЕМ КАРТЫ ПРЕДСКАЗАТЕЛЬНЫХ ПОТЕРЬ

В заключение мы покажем, что архитектуру системы из предыдущего раздела можно дополнительно улучшить, чтобы локализовать событие на изображениях, т. е. отметить с помощью ограничительной рамки место, где происходит событие. Механизм внимания, рассмотренный в разделе 12.4, генерирует только карту интенсивностей с размером сетки, зависящим от выходного пространственного разрешения кодировщика. Другими словами, карта внимания действует как указатель того, в каком месте кадра происходит действие; однако она не создает точную маску или ограничивающую рамку. Чтобы сформировать ограничивающую рамку, мы используем сеть прогнозирования регионов и карту предсказательных потерь для фильтрации этих прогнозов с использованием функции минимизации пространственно-временной энергии. Следующее описание взято из (Aakur, Sarkar, 2020).

Мы начинаем с извлечения релевантных признаков (раздел 12.5.1) из необработанных кадров в пространстве пикселей. Извлеченные признаки будут служить входными данными для сети прогнозирования регионов и модулей предсказания (раздел 12.5.2). Энергетическая функция объединяет предсказательную потерю и прогнозируемые регионы (ограничивающие рамки) для выделения *каналов действия* (action tube) (раздел 12.5.4), которые согласуются в пространстве и времени. На рис. 12.15 показаны четыре компонента архитектуры: (1) извлечение признаков и прогнозирование пространственной области, (2) самообучаемая модель предсказания будущего, (3) модуль обнаружения пространственно-временных ошибок и (4) процесс локализации действия на основе модуля обнаружения ошибок.

12.5.1. Извлечение признаков

Как и в предыдущих методах, для извлечения релевантных признаков используется кодировщик CNN. Однако извлеченные признаки в этом подходе используются в качестве входных данных для сети прогнозирования пространственной области и стека предсказаний. Прогноз области, по сути, представляет собой набор ограничивающих рамок, определяющих возмож-

ные области действия и соответствующие объекты для каждого кадра. Предварительно обученная CNN используется для извлечения признаков и генерации прогнозов.

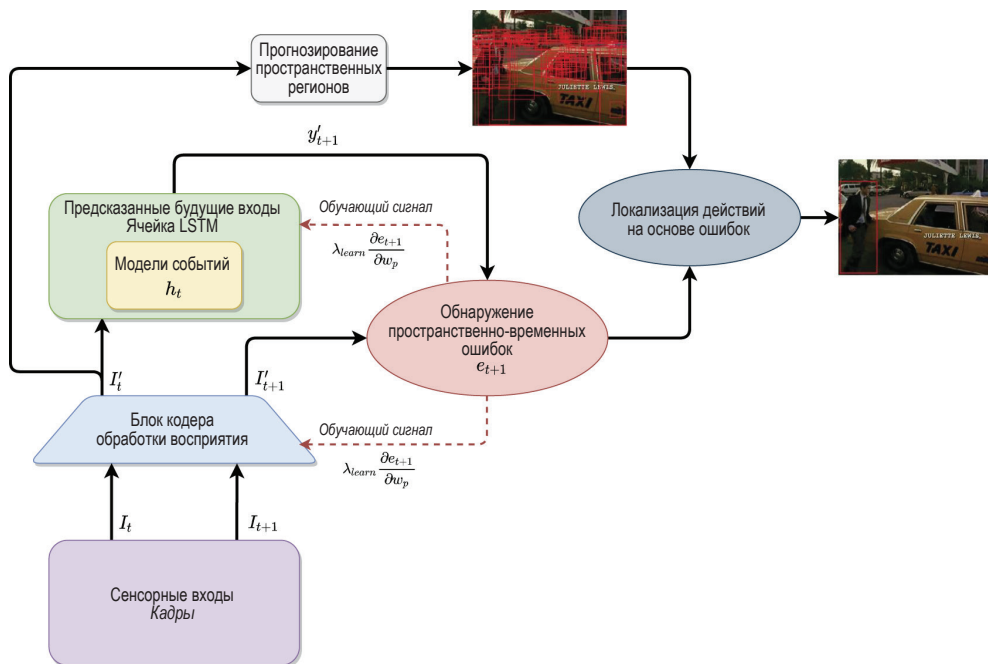


Рис. 12.15 ❖ Вариант 3: пространственно-временная локализация с использованием карты предсказательных потерь. Этот метод представляет собой вариант 2 с дополнительным компонентом, который локализует в пространстве кадра доминирующее действие на основе ошибки прогнозирования. Он состоит из четырех основных компонентов: (1) извлечение признаков и прогнозирование пространственной области, (2) самообучаемый фреймворк предсказания будущего, (3) модуль обнаружения пространственно-временных ошибок и (4) процесс локализации действия на основе модуля обнаружения ошибок

Мы используем прогнозы, не зависящие от класса (т. е. категория объекта игнорируется и учитываются только локализации на основе признаков) по двум основным причинам. Во-первых, мы не хотим делать никаких предположений о характеристиках актора, таких как метка, роль и аффорданс. Во-вторых, несмотря на значительный прогресс в обнаружении объектов, может быть много пропущенных обнаружений, особенно когда объект (или актер) выполняет действия, которые могут изменить его внешний вид. Следует отметить, что эти соображения могут привести к большому количеству прогнозов регионов, которые требуют аккуратного и тщательного отбора, но потенциально обеспечивают более высокие шансы на правильную локализацию.

12.5.2. Иерархический стек предсказания

Подобно подходам, упомянутым выше, текущая архитектура изучает предсказательную функцию на признаках высокого уровня, извлеченных кодировщиком. Сеть с LSTM применяется для прогнозирования следующего набора векторов признаков во временной последовательности, исходя из закодированного ввода и внутренней модели текущего события. Внутренняя модель эффективно фиксирует пространственно-временную динамику наблюдаемого события. Подобно предыдущей модели внимания, LSTM на каждом временном шаге обрабатывает набор векторов признаков, а не один вектор. Пространственное разрешение карт признаков определяется последним слоем свертки кодировщика.

Сеть LSTM выбрана не случайно; хотя другие методы, такие как сверточные декодеры (Jia et al., 2016) и модели смешанных сетей (Vondrick et al., 2016), являются жизнеспособными альтернативами для предсказания будущего, мы предлагаем использовать рекуррентные сети по следующим причинам. Во-первых, мы хотим моделировать временную динамику во всех кадрах наблюдаемого действия (или события). Во-вторых, LSTM допускают несколько возможных вариантов будущего и, следовательно, не будут усреднять результаты этих возможных вариантов будущего, как это может быть в случае с другими прогнозными моделями. Допустим, мы наблюдаем последовательность кадров $I_a = (I_a^1, I_a^2, \dots, I_a^n)$, соответствующих деятельности a . В случае видео со сложной структурой, например учебного видеоролика или спортивного репортажа, следующий набор кадров может представлять деятельность b или c с равными вероятностями, определяемыми как $I_b = (I_b^1, I_b^2, \dots, I_b^m)$ и $I_c = (I_c^1, I_c^2, \dots, I_c^k)$ соответственно. Использование полностью связанного или сверточного прогнозного модуля, вероятно, приведет к предсказанию признаков, которые, как правило, являются средним значением двух действий b и c , т. е. $I_{avg}^k = \frac{1}{2}(I_b^k + I_c^k)$ для времени k . Это нежелательный

результат, потому что предсказанные признаки могут быть либо маловероятным результатом, либо, что более вероятно, находиться за пределами правдоподобного многообразия представлений. Рекуррентные сети, такие как RNN и LSTM, допускают одновременное существование нескольких вариантов предсказанного будущего, которые возможны в момент времени $t + 1$ при условии наблюдения за кадрами до момента времени t . В-третьих, поскольку мы работаем с локализацией на основе ошибок, использование LSTM гарантирует, что процесс обучения суммирует пространственно-временную ошибку во времени и может давать все более качественные прогнозы, особенно для действий большей продолжительности.

В отличие от предыдущих подходов, модуль предсказания здесь состоит из стека сетей LSTM. Выход одной LSTM используется как вход для другой LSTM. Каждая LSTM в стеке имеет свои собственные параметры, определяющие различную внутреннюю модель события в зависимости от его положения в иерархии. Эта архитектура позволяет моделировать как пространственные, так и временные зависимости, поскольку каждый LSTM более высокого уровня действует как прогрессивный декодер, который реагирует на временные

зависимости, выделенные LSTM более низкого уровня. Первая сеть LSTM выделяет пространственную зависимость, которая распространяется вверх по стеку предсказания.

Обновленное скрытое состояние первого (нижнего) слоя LSTM (h_t^1) зависит от текущего наблюдения f_t^S , предыдущего скрытого состояния (h_{t-1}^1) и состояния памяти (m_{t-1}^1). Каждая из LSTM более высокого уровня принимает на свой первый слой выходные данные LSTM предшествующего уровня (h_t^{l-1}) и состояние памяти (m_t^{l-1}) и может быть определена как $(h_t^l, m_t^l) = \text{LSTM}(h_t^{l-1}, h_t^{l-1}, m_t^{l-1})$. Заметим, что это отличается от типичной иерархической модели LSTM (Song et al., 2017) тем, что на более высокие LSTM влияют выходные данные LSTM более низкого уровня *на текущем временном шаге*, а не на предыдущем. В совокупности модель событий W_e описывается обучаемыми параметрами и их соответствующими смещениями из иерархического стека LSTM.

Следовательно, верхний уровень стека предсказаний действует как декодер, целью которого является предсказание следующего признака f_{t+1}^S с учетом всех предыдущих предсказаний $\hat{f}_1^S, \hat{f}_2^S, \dots, \hat{f}_t^S$, модели событий W_e и текущего наблюдения f_t^S . Мы моделируем эту функцию прогнозирования как логарифмически-линейную модель, определяемую уравнением

$$\log p(\hat{f}_{t+1}^S | h_t^l) = \sum_{n=1}^t f(W_e, f_t^S) + \log Z(h_t), \quad (12.9)$$

где h_t^l – скрытое состояние LSTM l -го уровня в момент времени t , а $Z(h_t)$ – константа нормализации. Стек предсказания LSTM действует как генеративный процесс для прогнозирования будущих признаков.

12.5.3. Потеря предсказания

Карта внимания в этой архитектуре извлекается непосредственно из фактических предсказательных потерь (рис. 12.7) в верхней части стека предсказаний. Потеря предсказания является фактором качества сделанных предсказаний и относительного пространственного распределения ошибок предсказания. Взвешенная потеря движения из уравнения (12.8) используется для вычисления веса α_{ij} , связанного с каждым пространственным положением (i, j) в прогнозируемом признаке \hat{f}_{t+1}^S как

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{m=1}^{w_k} \sum_{n=1}^{h_k} \exp(e_{mn})}, \quad (12.10)$$

где e_{ij} представляет собой взвешенную ошибку предсказания в точке (i, j) (12.8). Это уравнение можно рассматривать как функцию $a(f_t^S, h_{t-1}^i)$ состояния самого верхнего LSTM и входного признака f_t^S в момент времени t . Полученная матрица представляет собой карту внимания на основе ошибок, которая позволяет нам локализовать ошибку прогноза в определенном пространственном местоположении. А средняя пространственная ошибка во времени $E(t)$ используется для локализации во времени.

12.5.4. Извлечение каналов действий

Действия извлекаются в виде каналов с использованием целевой функции энергии, которая подлежит минимизации. Модуль локализации действия получает в качестве входных данных поток прогнозов ограничивающей рамки (несколько прогнозов на кадр) и поток карт предсказательных потерь (по одной на кадр). Функция энергии предназначена для извлечения когерентных каналов действий. Она фильтрует прогнозы с использованием карты предсказательных потерь и возвращает набор прогнозов с наиболее высокой вероятностью локализации действия. Это достигается путем добавления компонента энергии каждому из прогнозов ограничивающей рамки (\mathcal{B}_{it}) в момент времени t и выбора верхних k ограничивающих рамок с наименьшей энергией в качестве наших окончательных прогнозов. Энергия ограничивающей рамки \mathcal{B}_{it} определяется уравнением

$$E(\mathcal{B}_{it}) = w_{\alpha}\phi(\alpha_{ij}, \mathcal{B}_{it}) + w_t\delta(\mathcal{B}_{it}, \{\mathcal{B}_{j,t-1}\}), \quad (12.11)$$

где $\phi(\cdot)$ – функция, которая возвращает значение, характеризующее расстояние между центром ограничивающей рамки и положением максимальной ошибки, $\delta(\cdot)$ – функция, возвращающая минимальное пространственное расстояние между текущей ограничивающей рамкой и ближайшей ограничивающей рамкой из предыдущего временного шага. Константы w_{α} и w_t являются масштабными коэффициентами. На рис. 12.18 показан пример извлеченных каналов действий для одного действия в потоке кадров.

12.5.5. Результаты

12.5.5.1. Данные

Для оценки нашего метода локализации действий мы используем три общедоступных набора данных.

UCF Sports (Rodriguez et al., 2008) – это набор данных, состоящий из 10 классов спортивных действий, таких как катание на коньках и поднятие тяжестей, собранных из спортивных трансляций. Это интересный набор данных, поскольку он имеет высокую концентрацию различных сцен и движений, что затрудняет локализацию и распознавание. Мы используем для оценки так называемые *сплиты* (103 обучающих и 47 тестовых видеороликов), как предложено в (Lan et al., 2011).

JHMDB (Jhuang et al., 2013) состоит из 21 класса действий и 928 обрезанных видео. Все видео аннотированы положением человеческих суставов в каждом кадре. Эталонная ограничивающая рамка для задачи локализации действия выбирается таким образом, чтобы она охватывала все суставы. Этот набор данных предлагает несколько усложнений, таких как увеличение количества фоновых помех, высокое сходство между классами, сложное движение (включая движение камеры) и частично закрытые объекты наблюдения. Мы представляем все результаты как среднее значение по всем трем сплитам.

THUMOS'13 (Jiang et al., 2014) – это подмножество набора данных UCF-101 (Soomro et al., 2012), состоящее из 24 классов и 3207 видео. Для каждого из классов задачи локализации действия предусмотрены эталонные ограничивающие рамки. Он также известен как набор данных UCF-101-24. Мы проводим эксперименты и сообщаем о результатах первого сплита в соответствии с предыдущими работами (Li et al., 2018; Soomro, Shah, 2017).

Мы также проанализировали способность метода обобщать видео, сфокусированное на персоне (эго-видео), оценивая его в *задаче прогнозирования взгляда* с обучением без учителя. В когнитивной психологии имеется достаточно доказательств сильной корреляции между точками, на которые направлен взгляд, и локализацией действия (Tipper et al., 1992). Поэтому задача предсказания взгляда представляется нам разумной мерой обобщения локализации действия в эго-видео. Мы оцениваем точность на наборе данных **GTEA Gaze** (Fathi et al., 2012), который состоит из 17 последовательностей задач, выполняемых 14 испытуемыми, причем каждая последовательность длится около 4 минут. Мы используем общепринятые сплиты наборов данных GTEA, определенные в предыдущих работах (Fathi et al., 2012).

12.5.5.2. Показатели и базовые уровни

В решении задачи локализации действия мы опирались на предыдущие исследования (Li et al., 2018; Soomro, Shah, 2017) и получили значения *именованной средней точности* (mean average precision, mAP) при различных порогах перекрытия, полученной путем вычисления пересечения поверх объединения (IoU) предсказанных и эталонных ограничивающих рамок. Мы также оцениваем качество прогнозов ограничивающей рамки, измеряя среднее значение IoU за кадр и полноту прогнозов рамки при различных коэффициентах перекрытия.

Поскольку наш метод основан на обучении без учителя, мы получаем метки классов путем кластеризации изученных представлений с использованием алгоритма k -средних. Хотя более сложная кластеризация может дать лучшие результаты распознавания (Soomro, Shah, 2017), алгоритм k -средних позволяет нам оценить робастность изученных признаков. Мы оцениваем наш метод по двум параметрам K_{gt} и K_{opt} , где первый – это количество кластеров, равное количеству классов эталонных действий, а второй – оптимальное количество, полученное с помощью *метода локтя* (Kodinariya, Makwana, 2013) соответственно. Из наших экспериментов следует, что K_{opt} в три раза превышает количество эталонных классов, что не лишено смысла и было рабочим предположением в других методах кластеризации, основанных на глубоком обучении (Hershey et al., 2016). Оценочное сравнение с эталонными кластерами выполнялось с использованием венгерского метода, как это делалось в предыдущих методах обучения без учителя (Ji et al., 2019; Xie et al., 2016). Для оценки эффективности нашего протокола обучения мы также провели сравнение с другими подходами – LSTM и на основе внимания (раздел 12.5.5.3.3).

Для задачи прогнозирования взгляда мы оцениваем разные подходы с использованием метода *площади под кривой* (area under curve, AUC), который

измеряет площадь под кривой на графиках значимости для истинно положительных показателей по сравнению с ложноположительными при различных пороговых значениях. Мы также определили *среднюю угловую ошибку* (average angular error, AAE), которая отражает угловое расстояние между прогнозируемым и реальным положением взгляда. Поскольку выход нашей модели представляет собой график значимости, AUC является более подходящей метрикой, чем средняя угловая ошибка AAE, которая требует знания определенных местоположений.

12.5.5.3. Количественная оценка

В этом разделе мы представляем количественную оценку нашего подхода к двум различным задачам, а именно к локализации действия и предсказанию взгляда. Для задачи локализации действия мы оцениваем наш метод по двум аспектам – качество прогнозов и пространственно-временная локализация.

12.5.5.3.1. Качество прогнозов локализации

Сначала мы оцениваем качество наших прогнозов локализации, предполагая идеальное предсказание класса. Это позволяет нам независимо оценивать качество локализации, выполненной в режиме самообучения. Полученные оценки представлены в табл. 12.5 и сравниваются с оценками при полном обучении с учителем, частичном обучении с учителем и обучении без учителя. Как следует из данных в таблице, предсказательный метод превосходит многие альтернативные методы, основанные на полном или частичном обучении. Метод APT (Van Gemert et al., 2015) достигает более высокой оценки локализации. Однако он выдает в среднем 1500 прогнозов на видео, тогда как наш подход возвращает примерно 10 прогнозов. Большое количество прогнозов локализации для каждого видео может привести к более высоким значениям полноты отклика и IoU, но усложняет задачу локализации, т. е. затрудняет маркировку действий для каждого видео, и может повлиять на возможность обобщения на другие домены (области деятельности).

Кроме того, следует отметить, что наш метод выдает прогнозы в потоковом режиме, в отличие от многих других подходов, которые создают каналы действий на основе движения, вычисляемого по всему видео. Во втором случае локализация действий в реальном времени в потоковом видео может быть затруднена.

12.5.5.3.2. Пространственно-временная локализация действия

Мы протестировали наш подход к задаче пространственно-временной локализации. Полученная оценка позволяет нам проанализировать робастность признаков, полученных посредством предсказания с самообучением. Мы генерируем метки классов на уровне видео с помощью кластеризации и используем стандартные метрики оценки (раздел 12.5.5.2) для количественной оценки точности. Кривые AUC относительно различных порогов перекрытия представлены на рис. 12.16. Мы сравниваем их с набором базовых значений для полного обучения с учителем, частичного обучения и обучения без учителя на всех трех наборах данных.

Таблица 12.5. Сравнение с методами полного и частичного обучения с учителем на базовом уровне локализации действий, не зависящем от класса, в наборе данных UCF Sports. Представлена средняя точность локализации каждого подхода, т. е. средний IoU

Разметка	Метод	Точность, %
Полная	STPD (Tran, Yuan, 2011)	44,6
	Поиск макс. пути (Tran, Yuan, 2012)	54,3
Слабая	Ma et al. (Ma et al., 2013)	44,6
	GBVS (Grundmann et al., 2010)	42,1
	Soomro et al. (Soomro, Shah, 2017)	47,7
Нет	IME Tublets (Jain et al., 2014)	51,5
	APT (Van Gemert et al., 2015)	63,7
	Предсказательный подход	55,7

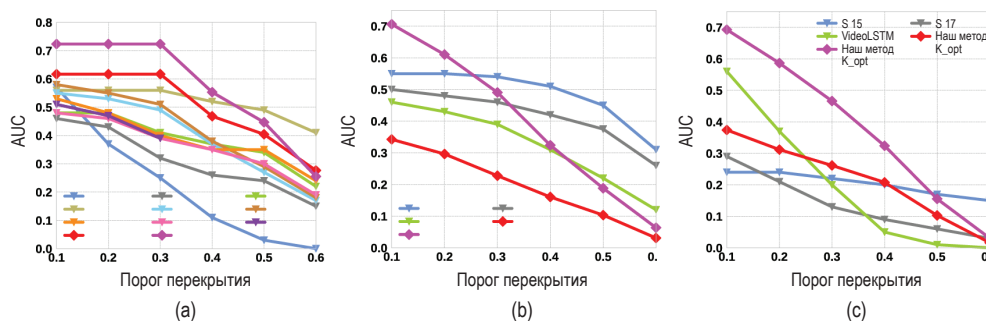


Рис. 12.16 ❖ Кривые AUC для задач локализации действия показаны для следующих наборов данных: (a) UCF Sports, (b) JHMDB и (c) THUMOS13. Мы сравниваем показатели нашего метода с базовыми данными при различных уровнях обучения, полученными из работ (Lan et al., 2011; Tian et al., 2013; Wang et al., 2014; Gkioxari, Malik, 2015; Jain et al., 2014; Soomro et al., 2015, 2016; Soomro, Shah, 2017; Hou et al., 2017) и VideoLSTM (Li et al., 2018)

В наборе данных UCF Sports (рис. 12.16a) наша архитектура превосходит все традиционные методы, включая несколько вариантов обучения с учителем, за исключением показателей метода (Gkioxari, Malik, 2015) при более высоких порогах перекрытия ($\sigma > 0,4$), когда мы устанавливаем число кластеров k равным числу эталонных классов. Когда допускаем некоторую пересегментацию и используем *оптимальное* количество кластеров, наш метод превосходит все альтернативные подходы до $\sigma > 0,5$.

Экспериментируя с набором данных JHMDB (рис. 12.16b), мы обнаружили, что хотя наш метод обеспечивает высокую полноту отклика даже при более строгих пороговых значениях (77,8 % при $\sigma > 0,5$), большое движение камеры и внутриклассовые вариации оказывают значительное влияние на точность классификации. Следовательно, mAP страдает, когда мы устанавливаем k равным числу эталонных классов. Когда мы выбираем оптимальное количество кластеров, наш метод превосходит альтернативные варианты при

более низких порогах (mAP при $\sigma < 0,5$). Следует отметить, что другой метод с обучением без учителя (Soomro et al. (2017)) использует прогнозы обнаружения объектов из базовой модели Faster R-CNN для оценки «человечности» прогноза. Это предположение делает подход предвзятым к локализации действий, ориентированных на человека, и снижает его способность обобщать действия акторов, не являющихся людьми. В свою очередь, мы не делаем никаких предположений о характеристиках актора, сцены или динамики движения.

В наборе данных THUMOS13 (рис. 12.16с) мы достигаем стабильного преимущества по сравнению с исходными примерами обучения без учителя и частичного обучения с учителем при $k = k_{gt}$ и самых современных показателей mAP при $k = k_{opt}$. Примечательно, что мы конкурируем (когда $k = k_{gt}$) с моделью частичного обучения на основе внимания VideoLSTM (Li et al., 2018), которая использует convLSTM для временного моделирования вместе с механизмом пространственного внимания на основе CNN. Следует отметить, что наш метод демонстрирует на THUMOS13 более высокую полноту отклика (0,47 при $\sigma = 0,4$ и 0,33 при $\sigma = 0,5$) при более высоких порогах, по сравнению с другими современными методами, и устойчивость алгоритма локализации на основе ошибок к внутриклассовой изменчивости и окклюзии.

Качество кластеризации. Поскольку наблюдается значительная разница в оценке mAP, когда мы меняем количество кластеров в k -средних, мы измерили *однородность* (или чистоту) кластеризации. Оценка однородности отражает «качество» кластера, измеряя, насколько хорошо кластер моделирует данный класс достоверности. Поскольку мы допускаем пересегментацию кластеров, когда устанавливаем k равным оптимальному количеству кластеров, это является важной мерой надежности признака. Более высокая однородность указывает на то, что учитываются вариации внутри класса, поскольку все точки данных в определенном кластере принадлежат к одному и тому же эталонному классу. Мы наблюдаем среднюю оценку однородности 74,56 %, когда k равно количеству эталонных классов, и 78,97 %, когда используем оптимальное количество кластеров. Как следует из результатов, несмотря на чрезмерную сегментацию, каждый из кластеров обычно моделирует один класс действий с высокой степенью совпадения.

12.5.5.3.3. Сравнение с другими подходами на основе LSTM

Мы также сравнили нашу модель с другими моделями на основе LSTM и внимания, чтобы подчеркнуть важность парадигмы самообучения. Поскольку системы на основе LSTM могут иметь очень похожие архитектуры, мы учитываем разные требования и характеристики, такие как уровень аннотации, необходимый для обучения, и количество прогнозов локализации, возвращаемых для каждого видео. Мы сравнили три подхода, схожих по духу с нашим, – ALSTM (Sharma et al., 2015), VideoLSTM (Li et al., 2018) и Actor Supervision (Escorcia et al., 2020) – и свели результаты в табл. 12.6. Видно, что наш подход значительно превосходит VideoLSTM и ALSTM на наборе данных THUMOS13 как по полноте отклика, так и по mAP при $\sigma = 0,2$.

Таблица 12.6. Сравнение нашей архитектуры с другими методами на основе LSTM и внимания на наборе данных THUMOS'13. Представлены средние значения полноты отклика при различных порогах перекрытия, mAP при пороге перекрытия 0,2 и среднем количестве прогнозов на кадр

Метод	Разметки		Число предложений	Средняя полнота отклика					mAP при $\sigma = 0,2$
	Метки	Рамки		0,1	0,2	0,3	0,4	0,5	
ALSTM (Sharma et al., 2015)	✓	✗	1	0,46	0,28	0,05	0,02	–	0,06
VideoLSTM (Li et al., 2018)	✓	✗	1	0,71	0,52	0,32	0,11	–	0,37
Actor Supervision (Escorcia et al., 2020)	✓	✗	~1000	0,89	–	–	–	0,44	0,46
Наш метод	✗	✗	~10	0,84	0,72	0,58	0,47	0,33	0,59

Модель Actor Supervision (Escorcia et al., 2020) превосходит наш вариант по отклику, но следует отметить, что прогнозы регионов зависят от двух факторов: (1) прогнозов акторов на основе обнаружения объектов и (2) механизма фильтрации, ограничивающего прогнозы, основанные на эталонных классах действий, которые могут увеличить требования к обучению и ограничить обобщаемость. Также следует иметь в виду, что возврат большего количества прогнозов локализации может увеличить отклик ценой ухудшения обобщения.

12.5.5.3.4. Вариативные исследования

Наша предсказательная модель имеет три основных компонента, которые больше всего влияют на ее качество: (1) модуль прогнозирования региона, (2) модуль предсказания будущего и (3) модуль локализации действий на основе ошибок. Мы рассматриваем и оцениваем несколько альтернатив всем трем модулям. Мы взяли выборочный поиск (Uijlings et al., 2013) и EdgeBox (Zhu et al., 2015) в качестве альтернативных методов прогнозирования региона для SSD.

Чтобы оценить эффективность использования локализации на основе ошибок, мы воспользовались методом локализации на основе внимания для локализации действия в качестве приближения ALSTM (Sharma et al., 2015). Мы также оценили одноуровневый предсказатель LSTM с полносвязной сетью декодеров (Aakur, Sarkar, 2019) для аппроксимации задачи локализации. Оценили эффект прогнозирования на основе внимания, вводя слой внимания Бахданау (Bahdanau et al., 2014) перед предсказанием в качестве альтернативы модулю локализации действий на основе ошибок.

Эти вариативные исследования проведены на наборе данных UCF Sports. Результаты представлены на рис. 12.17а. Можно видеть, что использование локализации на основе ошибок прогнозирования дает заметно лучший результат по сравнению с подходом к локализации на основе обученного внимания. Можно также заключить, что выбор методов прогнозирования области оказывает некоторое влияние на точность модели, при этом выборочный поиск и прогнозы EdgeBox работают немного лучше при более высоких порогах ($\sigma \in (0,4, 0,5)$) за счет увеличения времени вывода и дополнительных

прогнозов ограничивающей рамки (50 по сравнению с 10 из прогнозирования региона на основе SSD). Использование SSD для генерации прогнозов позволяет нам распределять веса между задачами кодирования кадров и прогнозирования регионов и уменьшать объем памяти и вычислительные ресурсы алгоритма. Мы также обнаружили, что использование внимания как части модуля прогнозирования значительно влияет на показатели качества архитектуры. Возможно, это можно отнести на счет целевой функции, которая направлена на минимизацию ошибки прогноза. Обратите внимание, что в данном случае для локализации событий мы используем внимание на основе ошибок, в отличие от изученного вектора внимания из раздела 12.4. Мы обнаружили, что использование внимания Бахданау для кодирования признаков может повлиять на функцию прогнозирования нашей модели.

12.5.5.3.5. Прогнозирование взгляда с обучением без учителя

Наконец, мы оцениваем способность обобщать персонализированное видео, количественно определяя точность модели в задаче прогнозирования взгляда с обучением без учителя. Поскольку нам не нужны никакие аннотации или другие вспомогательные данные, мы используем для этой задачи прежнюю архитектуру и стратегию обучения. Оцениваем набор данных взгляда GTEA и сравниваем его с другими моделями, обучаемыми без учителя, в табл. 12.7. Как видно из таблицы, нами получены конкурентоспособные результаты в задаче прогнозирования взгляда, превосходящие все сравнительные показатели как по AUC, так и по AAE. Следует отметить также, что мы превосходим метод смещения центра по показателю AUC. Метод смещения центра (center bias) использует пространственное смещение в персонализированных изображениях и всегда предсказывает центр видеокadra как прогнозируемое положение взгляда. Значительное улучшение показателя AUC говорит о том, что наш подход предсказывает фиксации взгляда, которые более близки к эталонным, чем в подходе со смещением центра. Учитывая, что модель не была специально разработана для этой задачи, это замечательный результат, особенно с учетом показателей моделей, основанных на традиционном обучении с учителем, таких как DFG (Zhang et al., 2017), которые достигают 10,6 и 88,3 для AUC и AAE соответственно.

Таблица 12.7. Сравнение задачи прогнозирования взгляда с обучением без учителя на наборе данных GTEA с современными моделями

	Itti, Koch (2000)	GBVS (Harel et al., 2007)	AWS-D (Leboran et al., 2016)	Смещение центра	OBDL (Sayed et al., 2015)	Наш метод
AUC	0,747	0,769	0,770	0,789	0,801	0,861
AAE	18,4	15,3	18,2	10,2	15,6	13,6

12.5.5.4. Качественная оценка

Мы обнаружили, что наш подход имеет стабильно высокий уровень полноты отклика для задачи локализации с разными наборами данных и доменами. Мы считаем, что действие правильно локализовано, если средний IoU по всем кадрам выше 0,5, то есть большинство (если не все) кадров в видео пра-

вильно локализованы. Значения полноты отклика и последующие значения AUC для каждого класса в наборе спортивных данных UCF представлены на рис. 12.17б и 12.7в. Для большинства классов (7 из 10, если точнее) у нас есть отклик более 80 % при пороге перекрытия 0,5. При визуальном анализе мы обнаруживаем, что пространственно-временная ошибка часто коррелирует с актором, но обычно не находится в центре наблюдаемой области и, таким образом, снижает качество выбранных прогнозов. Мы демонстрируем этот эффект на рис. 12.18. В первой строке показан входной кадр, во второй – внимание на основе ошибок, а в последней строке – окончательные прогнозы по локализации. Если генерируется больше прогнозов (как в случае с выборочным поиском и EdgeBox), мы можем получить более высокий отзыв (рис. 12.17б) и более высокий mAP.

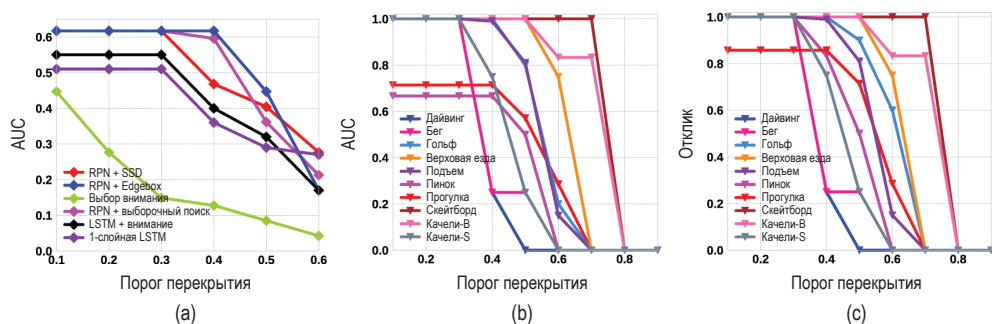


Рис. 12.17 ❖ Качественный анализ нашего метода на наборе данных UCF Sports: (а) абляционные вариации AUC, (б) AUC по классам и (с) полнота отклика ограничивающей рамки по классам при разных порогах перекрытия

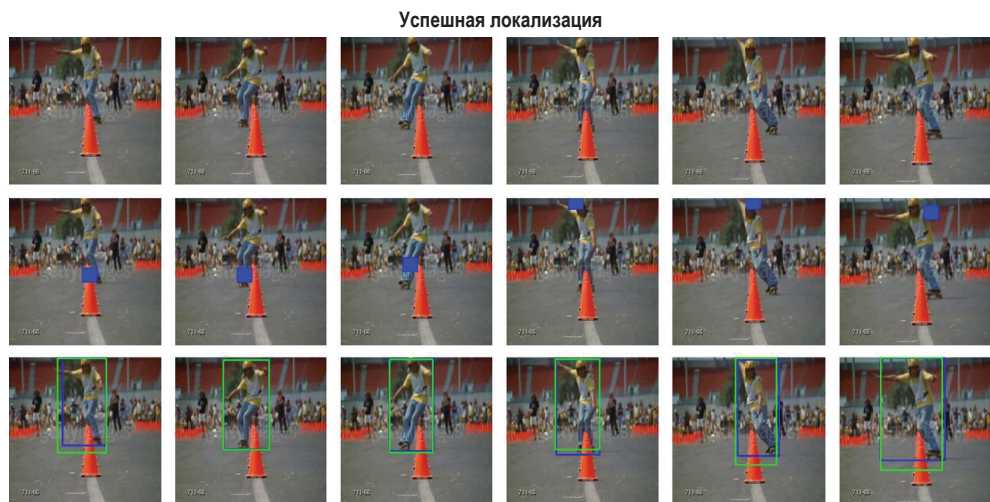


Рис. 12.18 ❖ Качественные примеры: локализация внимания на основе ошибок и окончательный прогноз. Зеленая рамка – предсказание; синяя рамка – эталон

12.6. ДРУГИЕ ПОДХОДЫ К СЕГМЕНТАЦИИ СОБЫТИЙ В КОМПЬЮТЕРНОМ ЗРЕНИИ

В настоящее время существуют три разных класса подходов к сегментации событий во времени, различаемых по критерию обучения: полное обучение с учителем, частичное обучение и обучение без учителя. Хотя традиционные подходы на основе полного обучения с учителем имеют более точную локализацию и лучшую точность маркировки, за это приходится платить очень большим количеством аннотаций. Потребность в размеченных обучающих данных плохо масштабируется с увеличением классов меток. Хотя подходы на основе частичного обучения (или с так называемой слабой разметкой) не требуют аннотаций на уровне кадра, в их основе лежит предположение о том, что существует большой аннотированный обучающий набор, который позволяет эффективно обнаруживать всех возможных акторов (как людей, так и предметы) в наборе классов действий. Подходы с обучением без учителя, такие как наш, не делают вышеупомянутых предположений, но могут привести к снижению точности локализации. Мы в некоторой степени решаем эту проблему, используя последние достижения в области механизмов прогнозирования регионов и робастных самообучающихся представлений. Один особый класс подхода с обучением без учителя – самообучение – использует для обучения только сами данные без аннотаций.

12.6.1. Методы на основе обучения с учителем

Методы, основанные на обучении с учителем, традиционно были доминирующим подходом к сегментации событий во времени. Они исходят из того, что для задачи есть полномасштабный учитель, и используют эталонные аннотации для выбора сегмента *с помощью классификации*, т. е. назначают метки семантически согласованным «фрагментам», чтобы сегментировать видео на его составные сегменты, при этом смежные кадры имеют одну и ту же метку. Общим приемом в этих подходах было извлечение признаков (либо созданных вручную, либо автоматизированных с использованием глубокого обучения) для применения машин опорных векторов на основе кадров (Kuehne et al., 2014) или моделирования временной динамики с использованием скрытых марковских моделей (Kuehne et al. et al., 2014), временных сверточных нейронных сетей (TCN) (Lea et al., 2017), пространственно-временных сверточных нейронных сетей (CNN) (Lea et al., 2016), рекуррентных сетей (Richard et al., 2017) и многих других. Хотя подходы на основе полноценного обучения с учителем привлекательны из-за их высокой точности, получение крупномасштабных аннотированных наборов данных, особенно с аннотациями на уровне кадра, может стоить довольно дорого и не всегда доступно. Эта проблема становится более заметной по мере увеличения детализации событий.

В некоторых подходах задачу локализации действия решают путем одно-временного создания прогнозов ограничивающей рамки и маркировки каж-

дой такой рамки прогнозируемым классом действия. Как генерация ограничивающей рамки, так и маркировка относятся к обучению с учителем, т. е. нужны обучающие эталоны как для ограничивающих рамок, так и для меток классов. В наиболее типичных методах при подготовке прогнозов используют достижения в области обнаружения объектов, чтобы добавить информацию о времени (Gkioxari, Malik, 2015; Hou et al., 2017; Jain et al., 2014; Soomro et al., 2015, 2016; Tian et al., 2013; Tran, Yuan, 2012; Wang et al., 2014). На последнем шаге обычно применяют алгоритм Витерби (Gkioxari, Malik, 2015) для связывания последовательности сгенерированных ограничивающих рамок во времени.

12.6.2. Методы на основе частичного обучения с учителем

Основная идея частичного обучения заключается в том, чтобы уменьшить потребность в прямой разметке за счет использования сопровождающих текстовых описаний или инструкций в качестве косвенных обучающих данных для изучения резко дискриминативных признаков. В области временной сегментации видео существуют два распространенных подхода к частичному обучению с учителем: (1) использование описаний или инструкций в качестве слабой разметки (Bojanowski et al., 2014; Ding, Xu, 2018; Alayrac et al., 2016; Malmaud et al., 2015) и (2) после неполной временной локализации действий для обучения и вывода (Huang et al., 2016; Richard et al., 2017). Хотя такие методы моделируют временные переходы с использованием RNN, они по-прежнему полагаются на принудительную семантику для сегментации действий и, следовательно, нуждаются в определенном объеме обучающих данных.

Упомянутые подходы применяют для локализации действия, чтобы уменьшить потребность в обширных аннотациях (Escorcia et al., 2020; Lan et al., 2011; Li et al., 2018; Sharma et al., 2015). Обычно им требуются только метки на уровне полного видео, и они используют обнаружение объектов для создания прогнозов ограничительной рамки. Следует отметить, что подходы со слабой разметкой также используют метки и характеристики на уровне объекта для управления процессом выбора ограничивающей рамки. В некоторых моделях (Escorcia et al., 2020) используют трекер на основе подобию для связывания ограничивающих рамок во времени, чтобы обеспечить непрерывность локализации.

12.6.3. Методы на основе обучения без учителя

При обучении без учителя отсутствуют внешние аннотированные обучающие данные, которые можно использовать для определения правильности вывода модели. Такие методы изучены в существенно меньшей степени по сравнению с двумя упомянутыми ранее. Основная идея заключается в использовании кластеризации и дискриминативных признаков (Bhatnagar et

al., 2017; Sener, Yao, 2018). Такие модели используют либо LSTM (Bhatnagar et al., 2017), либо обобщенную модель Маллоу (Sener, Yao, 2018). Гарсия и др. (Garcia et al., 2018) исследуют использование генеративной сети LSTM для сегментации последовательностей, как это делаем мы. Однако они работают только с грубым временным разрешением на протяжении жизни изображений, снятых с интервалом до 30 секунд. Последовательные изображения, снятые с таким промежутком, при изменении событий дают большую вариативность, что облегчает их различение. Кроме того, они применяют итеративный процесс обучения, которого у нас нет.

Некоторые модели не нуждаются ни в метках, ни в ограничивающих рамках. Два наиболее распространенных подхода заключаются в создании прогнозов действий с использованием (1) супервокселей (Jain et al., 2014; Soomro, Shah, 2017) и (2) кластеризующих траекторий движения (Van Gemert et al., 2015). Следует отметить, что Соомро и Шах (Soomro, Shah, 2017) также используют характеристики объекта для оценки «человечности» каждого супервокселя при выборе прогнозов ограничивающей рамки.

Наш подход, который мы представили в этой главе, относится к классу методов локализации действий с обучением без учителя. Наиболее близкими подходами (в отношении архитектуры и темы) являются VideoLSTM (Li et al., 2018) и Actor Supervision (Escorcia et al., 2020), которые используют внимание в процессе генерации прогнозов ограничивающей рамки, но требуют разметки на уровне видео. Наша модель не требует никаких меток или аннотаций ограничительной рамки для обучения.

12.6.4. Методы на основе самообучения

Одна из разновидностей обучения без учителя, привлекающая внимание исследователей, – это методы на основе самообучения, в которых используются обучающие данные, но без аннотаций. Существует два основных типа методов на основе самообучения: (1) сокрытие подмножества данных и попытка предсказать его, т. е. предсказать закрытую часть изображения объекта по видимой части или предсказать цвет из уровня серого (Zhang et al., 2016; Vondrick et al., 2018); (2) изменение входных данных и прогнозирование функции (параметров), вызвавших это изменение (Gidaris et al., 2018; Doversch et al., 2015; Misra et al., 2016; Fernando et al., 2017). Предсказание части входных данных или неизменной версии входных данных заставляет сеть изучать полезные семантические признаки из набора данных. Подходы на основе контрастного обучения (Chen et al., 2020; Grill et al., 2020; Caron et al., 2020) подпадают под вторую категорию, когда входные данные изменяются и сеть вынуждена изучать представление, которое максимизирует сходство между исходным и измененным вводами. Было обнаружено, что эти приемы позволяют изучать довольно надежные представления, которые затем можно применять при решении разных задач, например разметки, оценки движения и т. д.

В контексте видео самообучение сводилось в основном к предсказанию следующей пары кадров изображения (Srivastava et al., 2015; Mathieu et al.,

2015; Neverova et al., 2017; Lotter et al., 2016; Finn et al., 2016). Прогнозы на уровне признаков было предложено использовать для обнаружения одновременно встречающихся пространств визуальных признаков (Wang et al., 2014) и изучения представлений для лучшего распознавания (Liu et al., 2018; Li et al., 2017). Упомянутые подходы предлагают исследование контекста посредством совместного появления признаков во временных последовательностях для изучения либо множественной корреляции, такой как моделирование движения и внешнего вида для создания будущих кадров (Li et al., 2017), либо для изучения одновременно встречающихся понятий, таких как носы и глаза на лицах (Wang et al., 2014).

Также опубликовано много работ по прогнозированию будущей деятельности (Sun et al., 2019; Luc et al., 2017; Ma et al., 2016; Liang et al., 2019; Kitani et al., 2012; Fragkiadaki et al., 2015; Walker et al., 2014; Alahi et al., 2016; Santoro et al., 2017; Hamilton et al., 2017). Однако все эти подходы используют стандартную методику глубокого обучения на основе аннотированных данных.

Наш подход похож на методику прогнозирующего обучения по видео, предложенную Вондриком и др. (Vondrick et al., 2016), но для концепций более высокого уровня, а не только для меток действий. Исследования мозга и когнитивных функций убедительно свидетельствуют о том, что прогнозы играют важную роль в процессах усвоения мозгом новых понятий. Основываясь на результатах целого ряда нейробиологических экспериментов, Хоукинс и др. (Hawkins et al., 2016) также предложили повторяющуюся архитектуру слоев прямого распространения, предиктивной памяти и пулинга по времени, на которой они продемонстрировали обнаружение аномалий в одномерных сигналах. Хигер (Heeger, 2017) показал эффективность многоуровневой архитектуры со связями «снизу вверх» и «сверху вниз», но на основе прогнозов временной обработки сигналов. Лоттер и др. (Lotter et al., 2016) реализовали стек предиктивного кодирования, но для прогнозирования на уровне видеокадра.

12.7. Выводы

Эта глава основана на исследованиях когнитивной науки, направленных на определение задачи сегментации событий и разработку высокоэффективных алгоритмов компьютерного зрения для пространственно-временной сегментации событий в видео. Новые подходы не требуют ни аннотированных данных, ни многократных проходов по данным. Они могут обрабатывать *поточковые* видеоданные, одновременно изучая надежные представления для сегментации событий.

Основная идея состоит в том, чтобы использовать прогностическое обучение, в соответствии с теорией сегментации событий (EST) в когнитивной науке, для обнаружения границ событий. Как и в EST, блок обработки восприятия (стек CNN) отправляет извлеченные признаки из текущего кадра (обусловленные рабочей моделью событий) в блок предсказания (LSTM, стек LSTM, модуль прогнозирования ограничивающей рамки), который прогнозирует

будущие перцептивные признаки. Несовпадение между предсказанными признаками и фактическими, вычисленное на следующем шаге времени, генерирует сигнал ошибки предсказания. Высокая ошибка сигнализирует о границе события. Начиная с этого момента необходима новая модель событий, чтобы делать прогнозы на будущее. Ошибка предсказания запускает механизм стробирования для обновления рабочей модели событий (скрытых состояний в LSTM).

Обширные эксперименты в различных областях демонстрируют, что предсказательное (предвосхищающее) обучение может изучать надежные представления событий из немаркированных *поточковых* видеопоследовательностей только с одной эпохой обучения (однократное прохождение через видео). На представлениях событий основаны самые современные результаты в области сегментации событий во времени (раздел 12.3) и пространственно-временной локализации действий (раздел 12.5). При этом новые методы обеспечивают точность, сопоставимую с традиционными методами, которые основаны на обучении с учителем и требуют больших объемов аннотированных обучающих данных. Кроме того, мы показали, что система предсказательного обучения может обрабатывать и сегментировать потоковые видеоданные чрезвычайно большой продолжительности (раздел 12.4) со скоростью обработки, близкой к обработке в реальном времени. Предсказательное обучение помогает преодолеть растущую зависимость от обучающих данных и перейти к визуальному пониманию открытого мира. Мы надеемся, что полученные нами результаты стимулируют дальнейшие исследования в этом многообещающем направлении и избавят отрасль компьютерного зрения от постоянно растущей потребности в аннотированных данных.

БЛАГОДАРНОСТИ

Это исследование было частично поддержано грантами Национального научного фонда США CNS 1513126, IIS 1956050 и IIS 1955230.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Aakur Sathyanarayanan N., Sarkar Sudeep, 2019. A perceptual prediction framework for self-supervised event segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Aakur Sathyanarayanan N., Sarkar Sudeep, 2020. Action localization through continual predictive learning. arXiv preprint. arXiv:2003.12185.
- Aakur Sathyanarayanan, de Souza Fillipe D. M., Sarkar Sudeep, 2019. Going deeper with semantics: exploiting semantic contextualization for interpretation of human activity in videos. In: IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE. ActEV: Activities in Extended Video, 2019. <https://actev.nist.gov/>.

- Alahi Alexandre, Goel Kratarth, Ramanathan Vignesh, Robicquet Alexandre, Fei-Fei Li, Savarese Silvio*, 2016. Social lstm: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971.
- Alayrac Jean-Baptiste, Bojanowski Piotr, Agrawal Nishant, Sivic Josef, Laptev Ivan, Lacoste-Julien Simon*, 2016. Unsupervised learning from narrated instruction videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4575–4583.
- Bahdanau Dzmitry, Cho Kyunghyun, Bengio Yoshua*, 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint. arXiv:1409.0473.
- Bhatnagar Bharat Lal, Singh Suriya, Arora Chetan, Jawahar C. V., CVIT K. C. I. S.*, 2017. Unsupervised learning of deep feature representation for clustering ego-centric actions. In: International Joint Conference on Artificial Intelligence (IJCAI). AAAI Press, pp. 1447–1453.
- Bojanowski Piotr, Lajugie R'emi, Bach Francis, Laptev Ivan, Ponce Jean, Schmid Cordelia, Sivic Josef*, 2014. Weakly supervised action labeling in videos under ordering constraints. In: European Conference on Computer Vision (ECCV). Springer, pp. 628–643.
- Caron Mathilde, Misra Ishan, Mairal Julien, Goyal Priya, Bojanowski Piotr, Joulin Armand*, 2020. Unsupervised learning of visual features by contrasting cluster assignments. In: Advances in Neural Information Processing Systems, vol. 33.
- Chen Ting, Kornblith Simon, Norouzi Mohammad, Hinton Geoffrey*, 2020. A simple framework for contrastive learning of visual representations. arXiv preprint. arXiv:2002.05709.
- Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina*, 2018. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint. arXiv:1810.04805.
- Ding Li, Xu Chenliang*, 2018. Weakly-supervised action segmentation with iterative soft boundary assignment. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Doersch Carl, Gupta Abhinav, Efros Alexei A.*, 2015. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430.
- Escorcia Victor, Dao Cuong D., Jain Mihir, Ghanem Bernard, Snoek Cees*, 2020. Guess Where? Actor-Supervision for Spatiotemporal Action Localization. Computer Vision and Image Understanding, vol. 192, p. 102886.
- Fathi Alireza, Li Yin, Rehgh James M.*, 2012. Learning to recognize daily actions using gaze. In: European Conference on Computer Vision. Springer, pp. 314–327.
- Fernando Basura, Bilen Hakan, Gavves Efstratios, Gould Stephen*, 2017. Self-supervised video representation learning with odd-one-out networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3636–3645.
- Finn Chelsea, Goodfellow Ian J., Levine Sergey*, 2016. Unsupervised learning for physical interaction through video prediction. CoRR. arXiv:1605.07157.
- Fragkiadaki Katerina, Levine Sergey, Felsen Panna, Malik Jitendra*, 2015. Recurrent network models for human dynamics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4346–4354.

- Garcia del Molino Ana, Lim Joo-Hwee, Tan Ah-Hwee*, 2018. Predicting visual context for unsupervised event segmentation in continuous photo-streams. In: ACM Conference on Multimedia (ACM MM). ACM, pp. 10–17.
- Gidaris Spyros, Singh Praveer, Komodakis Nikos*, 2018. Unsupervised representation learning by predicting image rotations. arXiv:1803.07728 [cs.CV].
- Gkioxari Georgia, Malik Jitendra*, 2015. Finding action tubes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 759–768.
- Grill Jean-Bastien, et al.*, 2020. Bootstrap your own latent-a new approach to self-supervised learning. In: Advances in Neural Information Processing Systems, vol. 33.
- Grundmann Matthias, Kwatra Vivek, Han Mei, Essa Irfan*, 2010. Efficient hierarchical graph-based video segmentation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2141–2148.
- Hamilton William L., Ying Rex, Leskovec Jure*, 2017. Representation learning on graphs: methods and applications. arXiv preprint. arXiv:1709.05584.
- Hard Bridgette M., Tversky Barbara, Lang David S.*, 2006. Making sense of abstract events: building event schemas. *Memory & Cognition* 34 (6), 1221–1235.
- Harel Jonathan, Koch Christof, Perona Pietro*, 2007. Graph-based visual saliency. In: Advances in Neural Information Processing Systems, pp. 545–552.
- Hawkins Jeff, Ahmad Subutai*, 2016. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, vol. 10, p. 23.
- Hawkins Jeff, Blakeslee Sandra*, 2004. On Intelligence. Macmillan.
- Heeger David J.*, 2017. Theory of cortical function. *Proceedings of the National Academy of Sciences* 114 (8), 1773–1782.
- Hershey John R., Chen Zhuo, Le Roux Jonathan, Watanabe Shinji*, 2016. Deep clustering: discriminative embeddings for segmentation and separation. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 31–35.
- Hochreiter Sepp, Schmidhuber Jürgen*, 1997. Long short-term memory. *Neural Computation* 9 (8), 1735–1780.
- Sayed Hossein Khatoonabadi, Vasconcelos Nuno, Bajic Ivan V., Shan Yufeng*, 2015. How many bits does it take for a stimulus to be salient? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5501–5510.
- Hou Rui, Chen Chen, Shah Mubarak*, 2017. Tube convolutional neural network (T-CNN) for action detection in videos. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5822–5831.
- Huang De-An, Fei-Fei Li, Niebles Juan Carlos*, 2016. Connectionist temporal modeling for weakly supervised action labeling. In: European Conference on Computer Vision (ECCV). Springer, pp. 137–153.
- Itti Laurent, Koch Christof*, 2000. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research* 40 (10–12), 1489–1506.
- Jain Mihir, Van Gemert Jan, J'egou, Herv'e Bouthemy, Patrick Snoek, Cees G. M.*, 2014. Action localization with tubelets from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 740–747.
- Jhuang Hueihan, Gall Juergen, Zuffi Silvia, Schmid Cordelia, Black Michael J.*, 2013. Towards understanding action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3192–3199.

- Ji Xu, Henriques João F., Vedaldi Andrea*, 2019. Invariant information clustering for unsupervised image classification and segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874.
- Jia Xu, De Brabandere Bert, Tuytelaars Tinne, Gool Luc V.*, 2016. Dynamic filter networks. In: *Neural Information Processing Systems*, pp. 667–675.
- Jiang Yu-Gang, Liu Jingen, Roshan Zamir A., Toderici George, Laptev Ivan*, 2014. Mubarak Shah, and Rahul Sukthankar. THUMOS challenge: Action recognition with a large number of classes.
- Kitani Kris M., Ziebart Brian D., Bagnell James Andrew, Hebert Martial*, 2012. Activity forecasting. In: *European Conference on Computer Vision*. Springer, pp. 201–214.
- Kodinariya Trupti M., Makwana Prashant R.*, 2013. Review on determining number of cluster in k-means clustering. *International Journal* 1 (6), 90–95.
- Kuehne Hilde, Arslan Ali, Serre Thomas*, 2014. The language of actions: recovering the syntax and semantics of goal-directed human activities. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 780–787.
- Kuehne Hilde, Gall Juergen, Serre Thomas*, 2016. An end-to-end generative framework for video segmentation and recognition. In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 1–8.
- Kurby Christopher A., Zacks Jeffrey M.*, 2008. Segmentation in the perception and memory of events. *Trends in Cognitive Sciences* 12 (2), 72–79.
- Lan Tian, Wang Yang, Mori Greg*, 2011. Discriminative figure-centric models for joint action localization and recognition. In: *2011 International Conference on Computer Vision*. IEEE, pp. 2003–2010.
- Lea Colin, Reiter Austin, Vidal Ren'e, Hager Gregory D.*, 2016. Segmental spatiotemporal cnns for fine-grained action segmentation. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 36–52.
- Lea Colin, Flynn Michael D., Ren'e Vidal, Austin Reiter, Hager Gregory D.*, 2017. Temporal convolutional networks for action segmentation and detection. In: *IEEE International Conference on Computer Vision (ICCV)*.
- Leboran Victor, Garcia-Diaz Anton, Fdez-Vidal Xose R., Pardo Xose M.*, 2016. Dynamic whitening saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (5), 893–907.
- LeCun Yann, Bengio Yoshua, et al.*, 1995. Convolutional networks for images, speech, and time series.
- Lei Peng, Todorovic Sinisa*, 2018. Temporal deformable residual networks for action segmentation in videos. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6742–6751.
- Li Ruiyu, Tapaswi Makarand, Liao Renjie, Jia Jiaya, Urtasun Raquel, Fidler Sanja*, 2017. Situation recognition with graph neural networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4173–4182.
- Li Zhenyang, Gavriluk Kirill, Gavves Efstratios, Jain Mihir, Snoek Cees GM*, 2018. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding* 166, 41–50.
- Liang Junwei, Jiang Lu, Carlos Niebles Juan, Hauptmann Alexander G., Fei-Fei Li*, 2019. Peeking into the future: predicting future person activities and locations in videos. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5725–5734.

- Liu Wenqian, Sharma Abhishek, Camps Octavia, Szaiaer Mario*, 2018. DYAN: a dynamical atoms-based network for video prediction. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 170–185.
- Lotter William, Kreiman Gabriel, Cox David*, 2016. Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint. arXiv: 1605.08104.
- Luc Pauline, Neverova Natalia, Couprie Camille, Verbeek Jakob, LeCun Yann*, 2017. Predicting deeper into the future of semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 648–657.
- Luong Minh-Thang, Pham Hieu, Manning Christopher D.*, 2015. Effective approaches to attention-based neural machine translation. arXiv preprint. arXiv: 1508.04025.
- Ma Shugao, Sigal Leonid, Sclaroff Stan*, 2016. Learning activity progression in lstms for activity detection and early detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1942–1950.
- Ma Shugao, Zhang Jianming, Ikizler-Cinbis Nazli, Sclaroff Stan*, 2013. Action recognition and localization by hierarchical space-time segments. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2744–2751.
- van der Maaten Laurens, Hinton Geoffrey*, 2008. Visualizing data using t-SNE. Journal of Machine Learning Research 9, 2579–2605.
- Malmaud Jonathan, Huang Jonathan, Rathod Vivek, Johnston Nick, Rabinovich Andrew, Murphy Kevin*, 2015. What's cookin'? Interpreting cooking videos using text, speech and vision. arXiv preprint. arXiv:1503.01558.
- Mathieu Michaël, Couprie Camille, LeCun Yann*, 2015. Deep multi-scale video prediction beyond mean square error. CoRR. arXiv:1511.05440 [abs].
- Misra Ishan, Zitnick C. Lawrence, Hebert Martial*, 2016. Shuffle and learn: unsupervised learning using temporal order verification. In: European Conference on Computer Vision. Springer, pp. 527–544.
- Neverova Natalia, Luc Pauline, Couprie Camille, Verbeek Jakob J., LeCun Yann*, 2017. Predicting deeper into the future of semantic segmentation. CoRR abs. arXiv: 1703.07684.
- Radvansky Gabriel A., Zacks Jeffrey M.*, 2014. Event Cognition. Oxford University Press.
- Richard Alexander, Gall Juergen*, 2016. Temporal action detection using a statistical language model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3131–3140.
- Richard Alexander, Kuehne Hilde, Gall Juergen*, 2017. Weakly Supervised Action Learning with RNN Based Fineto-Coarse Modeling. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1.2, p. 3.
- Richmond Lauren L., Zacks Jeffrey M.*, 2017. Constructing experience: event models from perception to action. Trends in Cognitive Sciences 21 (12), 962–980.
- Rodriguez Mikel D., Ahmed Javed, Shah Mubarak*, 2008. Action Mach a spatio-temporal maximum average correlation height filter for action recognition. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.
- Russakovsky Olga, et al.*, 2015. Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV) 115 (3), 211–252.
- Santoro Adam, Raposo David, Barrett David G., Malinowski Mateusz, Pascanu Razvan, Battaglia Peter, Lillicrap Timothy*, 2017. A simple neural network module for

- relational reasoning. In: *Advances in Neural Information Processing Systems*, pp. 4967–4976.
- Sener Fadime, Yao Angela*, 2018. Unsupervised learning and segmentation of complex activities from video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Sharma Shikhar, Kiros Ryan, Salakhutdinov Ruslan*, 2015. Action Recognition Using Visual Attention. *Neural Information Processing Systems: Time Series Workshop*.
- Shipley Thomas F., Zacks Jeffrey M.*, 2008. *Understanding Events: From Perception to Action*, vol. 4. Oxford University Press.
- Simonyan Karen, Zisserman Andrew*, 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. arXiv:1409.1556.
- Song Jingkuan, Gao Lianli, Guo Zhao, Liu Wu, Zhang Dongxiang, Tao Shen Heng*, 2017. Hierarchical LSTM with adjusted temporal attention for video captioning. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, pp. 2737–2743.
- Soomro Khurram, Idrees Haroon, Shah Mubarak*, 2015. Action localization in videos through context walk. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3280–3288.
- Soomro Khurram, Idrees Haroon, Shah Mubarak*, 2016. Predicting the where and what of actors and actions through online action localization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2648–2657.
- Soomro Khurram, Shah Mubarak*, 2017. Unsupervised action discovery and localization in videos. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 696–705.
- Soomro Khurram, Zamir Amir Roshan, Shah Mubarak*, 2012. UCF101: a dataset of 101 human actions classes from videos in the wild. *arXiv preprint*. arXiv:1212.0402.
- de Souza Fillipe D. M., Sarkar Sudeep, Srivastava Anuj, Su Jingyong*, 2016. Spatially coherent interpretations of videos using pattern theory. *International Journal on Computer Vision (IJCV)*, 1–21.
- Srivastava Nitish, Mansimov Elman, Salakhutdinov Ruslan*, 2015. Unsupervised learning of video representations using LSTMs. *CoRR*. arXiv:1502.04681 [abs].
- Stein Sebastian, McKenna Stephen J.*, 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, pp. 729–738.
- Sun Chen, Shrivastava Abhinav, Vondrick Carl, Sukthankar Rahul, Murphy Kevin, Schmid Cordelia*, 2019. Relational action forecasting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 273–283.
- Thorpe Simon, Fize Denis, Marlot Catherine*, 1996. Speed of processing in the human visual system. *Nature* 381 (6582), 520.
- Tian Yicong, Sukthankar Rahul, Shah Mubarak*, 2013. Spatiotemporal deformable part models for action detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2642–2649.
- Tipper Steven P., Lortie Cathy, Baylis Gordon C.*, 1992. Selective reaching: Evidence for action-centered attention. *Journal of Experimental Psychology. Human Perception and Performance* 18 (4), 891.

- Tran Du, Yuan Junsong*, 2012. Max-margin structured output regression for spatio-temporal action localization. In: *Advances in Neural Information Processing Systems*, pp. 350–358.
- Tran Du, Yuan Junsong*, 2011. Optimal spatio-temporal path discovery for video event detection. In: *CVPR 2011. IEEE*, pp. 3321–3328.
- Uijlings Jasper R. R., Van De Sande, Koen E. A., Gevers Theo, Smeulders Arnold W. M.*, 2013. Selective search for object recognition. *International Journal of Computer Vision (IJCV)* 104 (2), 154–171.
- Van Gemert Jan C., Jain Mihir, Gati Ella, Snoek Cees G. M., et al.*, 2015. APT: action localization proposals from dense trajectories. In: *BMVC*, vol. 2, p. 4.
- Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Łukasz, Polosukhin Illia*, 2017. Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.
- Vondrick Carl, Pirsivash Hamed, Torralb, Antonio*, 2016. Anticipating visual representations from unlabeled video. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 98–106.
- Vondrick Carl, Shrivastava Abhinav, Fathi Alireza, Guadarrama Sergio, Murphy Kevin*, 2018. Tracking emerges by coloring videos. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 391–408.
- Walker Jacob, Gupta Abhinav, Hebert Martial*, 2014. Patch to the future: unsupervised visual prediction. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3302–3309.
- Wang Hongxing, Yuan Junsong, Wu Ying*, 2014a. Context-aware discovery of visual co-occurrence patterns. *IEEE Transactions on Image Processing* 23 (4), 1805–1819.
- Wang Limin, Qiao Yu, Tang Xiaoou*, 2014b. Video action detection with relational dynamic-poselets. In: *European Conference on Computer Vision*. Springer, pp. 565–580.
- Xie Junyuan, Girshick Ross, Farhadi Ali*, 2016. Unsupervised deep embedding for clustering analysis. In: *International Conference on Machine Learning (ICML)*, pp. 478–487.
- Xu Kelvin, Ba Jimmy, Kiros Ryan, Cho Kyunghyun, Courville Aaron, Salakhudinov Ruslan, Zemel Rich, Bengio Yoshua*, 2015. Show, attend and tell: neural image caption generation with visual attention. In: *International Conference on Machine Learning*, p. 2048. 2057.
- Yang Zhilin, Dai Zihang, Yang Yiming, Carbonell Jaime, Salakhutdinov Russ R., Le Quoc V.*, 2019. Xlnet: generalized autoregressive pretraining for language understanding. In: *Advances in Neural Information Processing Systems*, pp. 5754–5764.
- Zacks Jeffrey M., Braver, Todd S., Sheridan, Margaret A., Donaldson, David I., Snyder, Abraham Z., Ollinger, John M., Buckner, Randy L., Raichle, Marcus E.*, 2001a. Human brain activity time-locked to perceptual event boundaries. *Nature Neuroscience* 4 (6), 651–655.
- Zacks, Jeffrey M., Tversky Barbara, Iyer Gowri*, 2001b. Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology. General* 130 (1), 29.
- Zacks Jeffrey M., Magliano Joseph P.*, 2011. Film, narrative, and cognitive neuroscience. *Art and the Senses* 435, 454.

- Zacks Jeffrey M., Speer Nicole K., Swallow Khena M., Braver Todd S., Reynolds Jeremy R.*, 2007. Event perception: a mind-brain perspective. *Psychological Bulletin* 133 (2), 273.
- Zacks Jeffrey M., Tversky Barbara*, 2001. Event structure in perception and conception. *Psychological Bulletin* 127 (1), 3.
- Zhang Mengmi, Teck Ma Keng, Hwee Lim Joo, Zhao Qi, Feng Jiashi*, 2017. Deep future gaze: gaze anticipation on egocentric videos using adversarial networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4372–4381.
- Zhang Richard, Isola Phillip, Efros Alexei A.*, 2016. Colorful image colorization. *arXiv:1603.08511 [cs.CV]*.
- Zhu Gao, Porikli Fatih, Li Hongdong*, 2015. Tracking randomly moving objects on edge box proposals. *arXiv preprint. arXiv:1507.08085*.

ОБ АВТОРАХ ГЛАВЫ

Рами Мунир – доктор философии, сотрудник кафедры вычислительной техники и технологии Университета Южной Флориды (USF) в Тампе. Он получил степень бакалавра технических наук и степень магистра в области машиностроения USF в 2015 и 2018 гг. соответственно. Также получил диплом с отличием и награду «Выдающийся выпускник» в 2015 г. и сертификат выпускника курсов робототехники от USF в 2018 году. Является лауреатом премии «Ранние инновации» от корпорации Intel. Его исследовательские интересы включают изучение иерархических представлений объектов и событий на основе самообучения, реализацию перцептивных и когнитивных теорий с использованием вычислительных методов глубокого обучения и прогностических моделей.

Сатьянараянан Аакур – доцент кафедры информатики в Университете штата Оклахома. Получил степень бакалавра в области электроники и техники связи Университета Анны, Ченнаи, в 2013 г., а затем степень магистра в области информационных систем управления и докторскую степень в области компьютерных наук в Университете Южной Флориды, Тампа, в 2015 и 2019 гг. соответственно. Его исследовательские интересы включают самообучающиеся модели, механизмы визуального понимания сцены и приложения глубокого обучения в геномике.

Судип Саркар – профессор и заведующий кафедрой вычислительной техники и технологии, а также заместитель вице-президента по специальным программам Университета Южной Флориды в Тампе. Получил степень бакалавра технических наук Индийского технологического института, Канпур, и степень доктора философии в области электроники Университета штата Огайо. Имеет более чем 25-летний опыт работы в области компьютерного зрения, алгоритмов и систем распознавания образов, десять патентов США, а также публикует высокоцитируемые статьи в журналах и на конференциях. Является членом AAAS, IEEE, IAPR, AIMBE и NAI. Работал во многих журналах и в настоящее время является главным редактором рецензируемого научного журнала о распознавании образов.

Глава 13

Вероятностные методы обнаружения аномалий в данных временных рядов с использованием обученных моделей для мультимедийных самосознательных систем

Авторы главы:

Карло Регаццони, Али Краяни, Джулия Славик и Лучио Марченаро,
DITEN, Генуэзский университет, Генуя, Италия

Краткое содержание главы:

- в этой главе представлена архитектура модели для обнаружения аномалий и объясняется ее значение для самосознательных систем с инкрементным обучением;
- мы кратко рассмотрим современные методы обнаружения аномалий, проведем их сравнение и отметим наш вклад;
- самосознательный агент, работающий с мультимедийными сенсорными данными, может обнаруживать аномалии на иерархическом уровне и обрабатывать как мало-, так и многомерные данные;
- *обобщенные состояния* (generalized state, GS) строятся непосредственно из малоразмерных наблюдений. Напротив, вариационный автоэнкодер используется для вывода GS более низкой размерности из данных высокой размерности;

- *обобщенная динамическая байесовская сеть* (generalized dynamic bayesian network, GDBN) изучается из GS и используется для прогнозирования динамики системы на нескольких уровнях;
- прогностические и диагностические сообщения, проходящие внутри GDBN, позволяют вычислять *обобщенные ошибки* (generalized error) для обнаружения новых возникающих правил;
- мы проверяем предложенный подход, используя реальные мультисенсорные данные полуавтономного самосознательного автомобиля.

13.1. ВВЕДЕНИЕ

Идентификация аномальных экземпляров данных представляет собой актуальную задачу в различных областях исследований. Благодаря алгоритмам обнаружения аномалий камеры видеонаблюдения могут распознавать, когда происходят потенциально опасные или насильственные события, такие как взлом, нападения, вооруженные ограбления или дорожно-транспортные происшествия; находить подозрительные зоны на медицинских изображениях, таких как маммограммы или томографические снимки, и помогать врачам в диагностике опухолей; обнаруживать мошеннические операции с кредитными картами как отклонения от обычного профиля использования клиентами. В общем, методы обнаружения аномалий составляют необходимый компонент во всех приложениях, требующих выявления отклонений от известного набора правил или моделей. Здесь важно отметить, что аномалии связаны с моделью и не являются абсолютными: в случае видеонаблюдения исходная модель может описывать нормальные взаимодействия между покупателями в магазине; в случае с медицинской визуализацией она может выделять характерные черты здорового органа; в случае с кредитной картой она может отслеживать обычные схемы транзакций пользователя по кредитной карте. Аномальные данные соответствуют отклонению от этих изученных моделей и вместо этого подчиняются другим правилам. Поэтому очевидно, что обнаружение аномалий, обладающее множеством применений, имеет фундаментальное значение и для искусственных самосознательных систем, которые могут постоянно учиться на новых ситуациях, поскольку оно позволяет системе отличать новые ситуации от уже испытанных.

Самосознание (self-awareness) можно определить как «способность становиться объектом собственного внимания» (Morin, 2006). Оно достигается, когда агент фокусируется не только на внешней среде, но и на эволюции своего собственного состояния. Концепция самосознания обычно приписывается биологическим существам и целенаправленно изучается некоторыми учеными, такими как Хайкин и Фустер (Haykin, Fuster, 2014), Фристон и др. (Friston et al., 2014) и Дамасио (Damasio, 1999). В последних исследованиях эту концепцию начали переносить на искусственные агенты, такие как по-

луавтономные автомобили и дроны или когнитивные радиоканалы¹. Чтобы эти агенты одновременно осознавали свое окружение и свое внутреннее состояние, они должны быть снабжены сенсорами двух разных типов: (1) *экстероцептивными* сенсорами для достижения осведомленности о внешней ситуации, такими как внешние камеры; (2) *проприоцептивными* сенсорами для самосознания, такими как датчик положения рулевого колеса в транспортном средстве. Благодаря способности сознать окружающую ситуацию и свое состояние эти агенты могут распознавать опыт, с которым они сталкиваются, исходя из ранее приобретенного опыта; это знание составляет основу для принятия решения и выполнения действия, влияющего на окружающую среду через набор исполнительных механизмов.

Как недавно предложили в (Regazzoni et al., 2020), чтобы считаться самосознательным, агент должен обладать шестью основными возможностями: инициализация, вывод, обнаружение аномалий, создание модели и интерфейс с управлением. Самосознательный агент сначала инициализируется с предопределенной моделью (1) и может запомнить модель (2); далее он использует изученную модель, чтобы делать выводы о своем будущем состоянии и будущих изменениях в окружающей среде (3); он обнаруживает новые ситуации, когда они появляются (4), и изучает новую модель для их описания (5); наконец, он использует полученные знания для принятия решений и внесения изменений в окружающую среду (6). Одна функция следует за другой, и они образуют замкнутый цикл, поскольку этап (5) переходит в (2). Обнаружение аномалий – это важнейший навык самосознательных систем, поскольку он запускает процессы идентификации неизвестных правил и создания новой модели. В таком контексте важно показать, как мы рассматриваем данные временных рядов и как можно определить аномалию путем сравнения прогноза для следующего момента времени с фактическим наблюдением в этот момент времени.

В этой главе мы предлагаем метод обнаружения аномалий в данных временных рядах, который следует включить в описанный выше цикл самосознания. Метод использует обобщенные динамические байесовские сети (GDBN) и их возможности передачи сообщений для определения иерархии различных уровней аномалий. Его можно использовать для данных с низкой и высокой размерностями. В многомерном случае (например, изображения) для преобразования многомерных данных в вероятностное низкоразмерное скрытое пространство можно использовать генеративные модели, такие как вариационные автокодировщики (VAE) или генеративно-состязательные сети (GAN). В этой главе мы обсудим в качестве примера использование VAE для уменьшения размерности данных.

Остальная часть главы структурирована следующим образом: в разделе 13.2 представлены основные понятия о генеративных моделях, GDBN, вариационных автоэнкодерах и современные способы обнаружения аномалий в низкоразмерных и многомерных данных; в разделе 13.3 описан предлагаемый нами метод; в разделе 13.4 представлены результаты, полученные на

¹ Например, «умная» сеть передатчиков в сетях 6G, способная динамично оценивать окружающую обстановку, абонентскую нагрузку и собственное состояние и оптимально самонастраиваться под обстоятельства. – Прим. перев.

предлагаемых данных; раздел 13.5 завершает главу и предлагает некоторые идеи для дальнейших исследований.

13.2. БАЗОВЫЕ ПОНЯТИЯ И ТЕКУЩЕЕ ПОЛОЖЕНИЕ ДЕЛ

13.2.1. Генеративные модели

В контексте машинного обучения можно выделить два типа моделей: *дискриминативные* и *генеративные*. Дискриминативные модели изучают условную вероятность метки класса с учетом некоторых наблюдаемых входных данных – это вариант классификации. С другой стороны, генеративные модели могут изучать скрытое распределение данных, на которых они обучаются, и генерировать выборки из того же распределения. *Байесовские сети* (Bayesian networks, BN) представляют собой разновидность генеративной модели, факторизирующей совместное распределение данных с учетом условных вероятностей, зависящих от скрытых переменных модели. В свою очередь, динамические байесовские сети (dynamic Bayesian networks, DBN) выполняют эту факторизацию во времени.

При работе с многомерными данными, такими как изображения, наиболее часто используемыми генеративными моделями являются *вариационные автокодировщики* (variational autoencoders, VAE) (Kingma, Welling, 2014) и *генеративно-сопоставительные сети* (generative adversarial networks, GAN) (Goodfellow et al., 2014). VAE изучают распределение вероятностей данных *явно*, обычно предполагая распределение Гаусса или гауссовой смешанной модели. VAE, предполагающие гауссово распределение данных, кодируют каждую выборку данных через среднее значение и дисперсию. Делая выборки из изученного распределения, они могут генерировать точки данных, аналогичные тем, на которых они обучались. И наоборот, GAN изучают распределение данных *неявно* без определения каких-либо параметров. VAE и GAN могут дополнительно использоваться внутри DBN для изучения связи между многомерными данными (например, изображениями) и низкоразмерными латентными состояниями (т. е. так называемой *моделью наблюдения*) (Slavic et al., 2021). Следовательно, в нашей проблеме определения генеративной модели мы можем различать два случая: в первом дана модель наблюдения (например, случай радиоданных); во втором модель наблюдения должна быть изучена (например, случай видеоданных). В этой главе мы рассказываем о результатах как частичного, так и полного изучения.

13.2.2. Модели динамической байесовской сети (DBN)

Байесовские сети (Bayesian network, BN) – это модели направленного ациклического графа, в которых узлы графа представляют собой набор случайных

величин, а ребра кодируют конкретную факторизацию совместного распределения этого набора в один момент времени. Однако во многих реальных приложениях большинство событий обнаруживаются не на основе одного момента времени, а на основе нескольких наблюдений, которые приводят к определенному событию (Mihajlovic, Petkovic, 2001).

Динамическая байесовская сеть (DBN) является расширением BN, которое может моделировать динамические процессы и описывать эволюцию системы во времени на иерархических уровнях. DBN позволяют кодировать вероятностные зависимости и обратные связи между случайными величинами в разных временных интервалах. DBN обычно представлена двумя наборами параметров. Первый содержит количество узлов в каждом временном интервале и соответствующую топологию, а второй набор состоит из *условных распределений вероятностей* (conditional probability distributions, CPD), описанных ребрами сети. DBN обобщают линейные динамические системы, представляя скрытые и наблюдаемые состояния в виде случайных переменных состояния, образующих графовую структуру, которая определяет соответствующие условные зависимости и компактную параметризацию модели. DBN могут разлагать данные со сложной и нелинейной динамикой на сегменты, которые можно объяснить с помощью более простых динамических блоков. Для представления динамических блоков и объяснения их поведения при переключении и их зависимости как от наблюдений, так и от непрерывных скрытых состояний можно использовать специфический класс моделей DBN, известных как *линейные динамические системы с переключением* (switching linear dynamic system, SLDS) (Fox et al., 2011). Обучение DBN состоит из изучения параметров и изучения структуры. Первое – это процесс изучения распределений дискретных или непрерывных скрытых переменных в DBN, тогда как второе – это процесс использования данных для изучения связей (т. е. условных вероятностей) между случайными величинами в DBN. Изучение как параметров, так и структуры зависит от рассматриваемой модели в пространстве состояний. Представление процесса временного ряда в пространстве состояний определяет prior $P(X_t)$, функцию перехода состояния $P(X_t|X_{t-1})$ и функцию наблюдения $P(Z_t|X_t)$. *Скрытые марковские модели* (hidden Markov model, HMM) и *модели фильтра Калмана* (Kalman filter model, KFM) можно рассматривать как возможные способы представления моделей пространства состояний, закодированных в простой DBN, которая содержит одну скрытую переменную и один наблюдаемый узел на срез, как показано на рис. 13.1. Предполагается, что скрытые дискретные переменные в HMM имеют полиномиальное дискретное распределение и эволюционируют в соответствии с правилами перехода, параметризованными в модели. Дискретные переменные и соответствующая матрица перехода могут быть изучены с помощью алгоритма кластеризации наблюдаемых данных временных рядов. В случае KFM эволюция перехода состояния становится линейно-гауссовой, например:

$$X_t = AX_{t-1} + BU_t + w_t, \quad (13.1)$$

где U_t – управляющий вход, реализующий родительский узел соответствующего X_t . В байесовской архитектуре обучение начинается с априорных зна-

ний о структуре модели (т. е. связях в DBN) и параметрах модели. В случае простой DBN (рис. 13.1) начальные знания о скрытой переменной состояния X_t с учетом распределения вероятностей $P(X_t)$ могут быть обновлены с использованием данных Z_t и использованы для вычисления апостериорной вероятности $P(X_t|Z_t)$ в соответствии с правилом Байеса:

$$P(X_t|Z_t) = \frac{P(Z_t|X_t)P(X_t)}{P(Z_t)}. \quad (13.2)$$

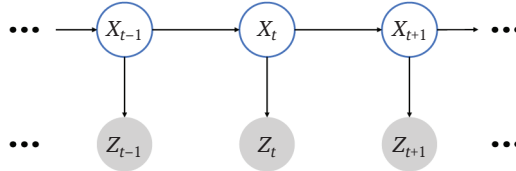


Рис. 13.1 ❖ Простая динамическая байесовская сеть. Сеть является скрытой марковской моделью (HMM), когда X является дискретной переменной, и моделью фильтра Калмана (KFM), когда X – непрерывная переменная

Возможная комбинация *фильтра частиц* (particle filter, PF) и *фильтра Калмана* (Kalman filter, KF), представленная в (Baydoun et al., 2018), может быть реализована на DBN как алгоритм вероятностного вывода. В байесовской архитектуре существуют различные типы вероятностных выводов, а именно нисходящий (или прогнозирующий) вывод и восходящий (или диагностический) вывод. Каждый из этих выводов основан на разложении требуемых вычислений на локальные вычисления в каждом узле сети, для чего требуется только передача сообщений с использованием алгоритма передачи сообщений (Winn, Bishop, 2005) по ребрам, связанным с этим узлом.

В данной главе предлагаемый нами подход обогащен расширенными возможностями по сравнению с классическим байесовским выводом, чтобы использовать прогностические и диагностические сообщения, поступающие в общий узел DBN, для расчета измерений аномалий на иерархических уровнях с использованием соответствующих вероятностных расстояний. Такие аномалии также можно использовать для постепенного изучения новых моделей, описывающих вариации на разных уровнях абстракции.

13.2.3. Вариационный автокодировщик

Как сказано в разделе 13.2.1, VAE – это генеративная модель. В простейшей базовой версии она состоит из кодировщика $Q_\theta(X|Z)$ и декодера $P_\phi(Z|X)$, которые можно построить в процессе обучения VAE на обучающих данных. Через θ и ϕ мы определяем параметры кодировщика и декодера соответственно. Кодировщик $Q_\theta(X|Z)$ позволяет представить каждую введенную в него выборку Z через два признака «бутылочного горла» – среднее значение μ и дисперсию σ^2 . С другой стороны, декодер $P_\phi(Z|X)$ синтезирует наблюдение Z из

скрытого состояния X , выбранного из $\mathcal{N}(\mu, \sigma^2)$. Следовательно, кодировщик позволяет уменьшить размерность наблюдений (т. е. данных изображения) и обрабатывать обнаружение аномалий на низкоразмерном уровне состояния; и наоборот, декодер позволяет снова получать многомерные данные и обрабатывать обнаружение аномалий на уровне наблюдения. Заметим, что в архитектуре DBN, описанной в предыдущем разделе, VAE можно рассматривать как модель наблюдения $P(Z_t|X_t)$.

Обучение VAE направлено на оптимизацию параметров θ и ϕ путем максимизации суммы нижней границы предельной вероятности каждого наблюдения Z набора данных D , как описано в (Kingma, Welling, 2014, 2019):

$$\mathcal{L}_{\phi, \theta}(D) = \sum_{Z \in D} \mathcal{L}_{\phi, \theta}(Z), \quad (13.3)$$

где $\mathcal{L}_{\phi, \theta}(Z)$ определяется как

$$\mathcal{L}_{\phi, \theta}(Z) = -D_{KL}(Q_{\theta}(X|Z) \parallel P_{\phi}(X) + E_{Q_{\theta}(X|Z)}[\log p_{\phi}(Z|X)], \quad (13.4)$$

где член D_{KL} – дивергенция KL. Таким образом, первый член измеряет разницу между распределением кодировщика $Q_{\theta}(X|Z)$ и априорным распределением $P_{\phi}(X)$, которое обычно представляет собой стандартное нормальное распределение $\mathcal{N}(0, 1)$. Второй член представляет собой ожидаемое логарифмическое правдоподобие наблюдения Z и заставляет VAE восстанавливать входные данные.

В данной работе используется способность VAE кодировать входную информацию в пространстве значительно меньшей размерности, проявляющем вероятностные свойства. Как отмечалось в разделе 13.2.1, VAE могут быть интегрированы в модель DBN в случае, когда модель наблюдения не задана, но также должна быть изучена.

13.2.4. Типы аномалий и методы обнаружения аномалий

Как было отмечено в разделе 13.1, аномалиями являются любые отклонения, проявляющиеся по отношению к эталонной модели. Понятие аномалии весьма обширно и зависит от типа данных, мощности взаимосвязи между задействованными переменными, структуры и распределения данных (Foorthuis, 2020). Различные типы аномалий – и, следовательно, расстояния до аномалий и алгоритмы обнаружения аномалий – могут быть обнаружены при работе с приложениями, не зависящими или зависящими от времени, с данными низкой размерности, такими как траектории, или многомерными данными, такими как изображения или графовые структуры и т. д. Чтобы оценить сложность и обширность проблематики обнаружения аномалий, мы снова обращаемся к работе (Foorthuis, 2020), в которой посредством обзора публикаций по обнаружению аномалий выделяются три широкие группы аномалий, 9 основных типов и 63 подтипа. Взяв в качестве примера видеоданные, Рамачандра и др. (Ramachandra et al., 2020) выделяют пять основных

типов аномалий: 1) аномалия внешнего вида; 2) аномалия кратковременного движения; 3) аномалия длительного движения; 4) групповая аномалия; 5) аномалия времени суток. Например, автомобиль, движущийся по улице, может либо увидеть аномальный объект в виде дорожного конуса в центре улицы (1), либо обнаружить аномальное движение в виде резкого торможения впереди идущего автомобиля (2), либо выполнить последовательность действий: аномальные движения, например, из-за того, что водитель заснул (3), наблюдение за двумя другими транспортными средствами, взаимодействующими ненормальным образом, как при столкновении (4); наконец, скорость движения, нормальная для светлого времени суток, не разрешена после захода солнца (5).

Работа Чандолы и др. (Chandola et al., 2009) формирует основу для классификации методов обнаружения аномалий. В этом разделе мы приводим краткий обзор определенных классов точечных аномалий. Авторы упомянутой статьи различают шесть общих методов обнаружения точечных аномалий; первые пять являются детерминированными методами, а шестой объединяет вероятностные методы. Далее мы кратко рассмотрим пять из шести этих методов. Кроме того, авторы указывают, как можно выделить дополнительные методы для контекстуальных аномалий, таких как аномальные последовательности во временном ряду.

Методы на основе классификации (CLA)

В методах этого типа изучается классификатор, который может либо отличать нормальные данные от аномальных на основе дискриминационной границы (одноклассовый классификатор), либо отличать классы нормальных данных друг от друга (многоклассовый классификатор). Во втором случае экземпляр определяется как аномальный, когда его нельзя с достаточной уверенностью связать ни с одним из нормальных классов. Любой алгоритм, обычно используемый для классификации, может быть адаптирован для этого типа метода, включая нейронные сети, байесовские сети, *машины опорных векторов* (support vector machines, SVM) и методы классификации на основе правил.

Методы на основе расстояния (DB)

Методы этого типа основаны на вычислении расстояния от точек данных до их ближайших соседей с использованием допущений о том, что нормальные данные встречаются в плотных окрестностях, тогда как аномалии находятся далеко от своих ближайших соседей. Поэтому выбор метрики расстояния имеет важное значение. В случае окрестностей с различной плотностью значение расстояния часто зависит от расстояний между точками в ближайшей окрестности.

Методы на основе кластеризации (CLU)

Эти методы основаны на предположениях, зависящих от выбранных методов кластеризации: 1) такие методы, как DBSCAN, ROCK или SNN, предполагают, что аномалии не относятся ни к одному из построенных кластеров; 2) при использовании таких методов, как SOM, предполагается, что аномальные

данные лежат далеко от центроида кластера; 3) наконец, можно использовать предположение, что нормальные данные относятся к большим кластерам, а аномальные – к малым кластерам.

Теоретико-информационные методы (IT)

В этих методах теоретико-информационные меры, такие как сложность Колмогорова, энтропия или относительная энтропия, используются для анализа информационного содержания данных и извлечения наиболее аномального подмножества точек данных (т. е. последовательности, подграфа, области изображения) относительно остальных данных.

Статистические/вероятностные методы (STA)

Методы, основанные на вероятностях, нацелены на моделирование нормального распределения признаков выборки, связанных с обучающими данными, тогда как аномалии могут быть обнаружены как *статистические выбросы* (outlier), т. е. выборки с низкой вероятностью (Rivera et al., 2020). Вероятность того, что выборка данных принадлежит определенному распределению, можно оценить с помощью вероятностных расстояний, таких как расхождение Бхаттачарии, Махаланобиса, Хеллингера или Кульбака–Лейблера. Обнаружение аномалий с использованием вероятностных (статистических) моделей можно разделить на две основные группы: параметрические, такие как методы *гауссовой модели смешения* (Gaussian mixture model, GMM) и методы регрессии, и непараметрические, такие как методы гистограммы или *оценки плотности ядра* (kernel density estimation, KDE) (Wang et al., 2019). Существенным преимуществом вероятностного подхода является то, что его можно легко обобщить на различные типы данных и модальности; недостаток связан с подгонкой данных к определенному распределению, что может быть неуместным в некоторых случаях (Aggarwal, 2016).

Кроме того, в публикации (Chandola et al., 2009) авторы определяют дополнительный набор методов обнаружения аномалий, а именно методы обнаружения спектральных аномалий, основанные на предположении, что данные, особенно многомерные, могут быть перенесены в подпространство более низкого измерения, где нормальные и аномальные данные легче различить. Примерами таких методов могут быть использование PCA для данных изображения или представление графов как матриц смежности. Эти методы можно использовать в качестве этапа предварительной обработки для других методов, определенных выше. Мы более подробно поговорим о проблеме уменьшения размерности в разделе 13.2.6, поэтому пока проигнорируем эту группу методов. Точно так же при обработке аномальных последовательностей данных, таких как данные временных рядов, обнаружение может быть выполнено либо путем адаптации предыдущих методов к контекстным аномалиям, либо с помощью методов, которые изучают структуру данных, например выполняя прогнозирование последующих моментов времени, исходя из предыдущих (*методы предсказания* (PRE)).

В табл. 13.1 представлена разработанная нами классификация с указанием основных характеристик для каждого метода. Обратите внимание, что все предлагаемые методы применимы к данным временных рядов и использу-

ют обучение без учителя или с частичным привлечением учителя, поэтому не нуждаются в полностью размеченных обучающих данных. Большинство методов также могут быть применены в архитектурах инкрементного обучения с использованием оценок аномалий, но эта проблема редко решается на современном уровне развития технологий. Классификация не дает оценки аномалии с доверительным интервалом, так как обычно в этом классе методов объект описывается только бинарно – как принадлежащий или не принадлежащий классу; теоретико-информационные методы также не имеют такой возможности. Кроме того, методы, основанные на расстоянии, и теоретико-информационные методы трудно распространить на данные высокой размерности. К вероятностным моделям можно применять только методы, основанные на прогнозировании, и статистические методы. Наконец, наш метод обладает всеми вышеупомянутыми характеристиками и дополнительно обеспечивает оценку аномалий на разных уровнях абстракции.

Таблица 13.1. Сравнение различных методов. * – не требуется для одноклассовой версии, необходим для многоклассовой версии; ** – подходит для большинства методов этой категории, но не для всех

	CLA	DB	CLU	IT	STA	PRE	Наш
Маркировка данных не нужна	*	✓	✓	✓	✓	✓	✓
Дает оценку аномалии с доверительным интервалом	х**	✓	✓	х	✓	✓	✓
Расширяемость до данных HD	✓	х	✓	х	✓	✓	✓
Применимость к вероятностным моделям	х	х	х	х	✓	✓	✓
Применимость к данным временных рядов	✓	✓	✓	✓	✓	✓	✓
Позволяет инкрементное обучение	✓	✓	✓	х	✓	✓	✓
Дает аномалию на разных уровнях абстракции	х	х	х	х	х	х	✓

13.2.5. Обнаружение аномалий в данных низкой размерности

В этой главе мы сосредоточимся на вероятностных методах с частичным обучением, основанных на представлениях DBN для данных временных рядов. В нескольких работах для обнаружения аномалий в различных областях использовались BN и DBN (Mascaro et al., 2014; Bronstein et al., 2001; Salotti, 2018). Этот тип сетей, обеспечивающий иерархическое вероятностное представление мира, особенно хорошо подходит для автономных систем, которые пытаются отражать человеческие рассуждения. BN дополнительно позволяют построить иерархию аномалий.

В частности, *марковский скачкообразный фильтр частиц* (Markov jump particle filter, MJPF) использовался в качестве базового байесовского фильтра, применяемого к DBN для обнаружения аномалий низкоразмерных данных, поступающих от различных типов датчиков мультимодальных физиче-

ских агентов, таких как полуавтономные транспортные средства, в качестве данных одометрии в (Baydoun et al., 2018) и контрольной информации в (Knapram et al., 2019).

13.2.6. Обнаружение аномалий в многомерных данных

Поскольку видеоданные являются многомерными, большинство алгоритмов обнаружения аномалий, разработанных для данных низкой размерности, включая MJPF, не могут выполнить эффективную попиксельную обработку видеопотока. Вместо этого необходимо сначала извлечь признаки из видеоданных, что позволит перенести задачу в пространство меньшей размерности (Chong, Tay, 2015). Процесс извлечения признаков может быть выполнен либо вручную, либо с помощью методов, основанных на глубоком обучении (deep learning, DL). В первом случае человеческие знания о конкретных проблемах, таких как окклюзия, изменения ракурса, масштаба или освещения, используются для разработки экстракторов признаков; одним из наиболее известных примеров является *гистограмма ориентированных градиентов* (histogram of oriented gradients, HOG) (Dalal and Triggs, 2005). В методах, основанных на DL, нейронные сети обучаются извлекать из данных соответствующие признаки для конкретной задачи путем минимизации функции потерь. Поскольку методы на основе DL в значительной степени превосходят ручные методы (Antipov et al., 2015; Alshazly et al., 2019; Nugroho, 2018), в этой главе мы рассматриваем только первые.

После процедуры уменьшения размерности к извлеченным признакам можно применить различные методы обнаружения аномалий, описанные в разделе 13.2.4, как показано в табл. 13.1. Кроме того, при использовании экстрактора признаков, который также позволяет выполнять восстановление исходных кадров из низкоразмерных признаков, обычно также используются методы обнаружения аномалий на основе восстановления. Как показано в (Kiran et al., 2018), методы этого типа учатся восстанавливать нормальные кадры или видеопоследовательности и определять плохо восстановленные образцы как аномалии; они обычно используют такие методы, как анализ основных компонентов или автокодировщики. Генеративные модели, такие как VAE и GAN, можно применять аналогичным образом, дополнительно получая возможность изучения основного распределения данных на уровне наблюдения.

Эта способность VAE и GAN соответствует идеям данной главы, поскольку в ней основное внимание уделяется разработке вероятностного метода, который может быть включен в архитектуру DBN для выявления аномалий и их использования для непрерывного обучения моделей на данных временных рядов. Благодаря уменьшению размерности наблюдаемых кадров VAE и GAN можно использовать в качестве нелинейных моделей наблюдения в структуре DBN, описанной в разделе 13.2.2.

Поскольку VAE могут явно описывать распределение вероятностей базовых данных, в нескольких работах предпринимались попытки использовать их

при создании модели линейной коммутации для данных высокой размерности (Watter et al., 2015; Johnson et al., 2016; Fraccaro et al., 2017; Becker-Ehmck et al., 2019). Однако эти работы не рассматривали обнаружение аномалий и непрерывное обучение моделей в качестве цели. Кроме того, эксперименты до сих пор проводились на очень простых наборах данных, отображающих только основные движения и наблюдения со статической точки зрения.

В другой работе (Ravanbakhsh et al., 2020) вместо этого были объединены GAN и DBN, реализующие одновременно обнаружение аномалий и непрерывное обучение. В этой работе разработана иерархия связанных кросс-модальных сетей GAN. Когда из-за обнаружения аномальных наблюдений или аномальной динамики возникает высокая аномалия, строится новая GAN для изучения новой ситуации. Вместо MJPF разрабатывается модель коммутации, состоящая из иерархии GAN на непрерывном уровне и HMM на дискретном уровне. Однако из-за изучения вероятности данных без явной параметризации – как это свойственно GAN – этот метод имеет ограничения и не предусматривает возможность использования некоторых расстояний до аномалий, обычно применяемых для статистических методов обнаружения аномалий (раздел 13.2.4).

Наиболее близкими к идеям данной главы являются работы (Slavic et al., 2021, 2020) и (Campo et al., 2020), в которых изучено сочетание VAE и MJPF, созданное для обнаружения аномалий и непрерывного обучения моделей.

13.3. АРХИТЕКТУРА ВЫЧИСЛЕНИЯ АНОМАЛИИ В САМОСОЗНАТЕЛЬНЫХ СИСТЕМАХ

13.3.1. Общее описание архитектуры

Предлагаемая нами архитектура изображена на рис. 13.2. На вход могут быть поданы данные низкой и высокой размерностей. После выполнения предварительной обработки данных низкой размерности и выделения признаков с помощью кодировщиков VAE из многомерных данных строится вектор обобщенного состояния, включающий отфильтрованные состояния как обобщенные ошибки, полученные из исходной модели, которая активируется во время вывода в реальном времени во время начальной итерации. В этом случае исходная модель представляет собой статические правила окружающей среды (т. е. ожидаемые динамические изменения практически равны нулю и на них влияют только случайные возмущения). Путем кластеризации обобщенных ошибок обучается модель обобщенной динамической байесовской сети (GDBN). При получении новых данных обученная модель используется для выполнения вывода через GDBN и извлекаются различные уровни аномалий (т. е. аномалия кадра, состояния, дискретного уровня). Если обнаруживается высокий уровень аномалии, соответствующие обобщенные состояния и обобщенные ошибки сохраняются, а новая модель строится для использования параллельно с предыдущими в следующем выводе GDBN.

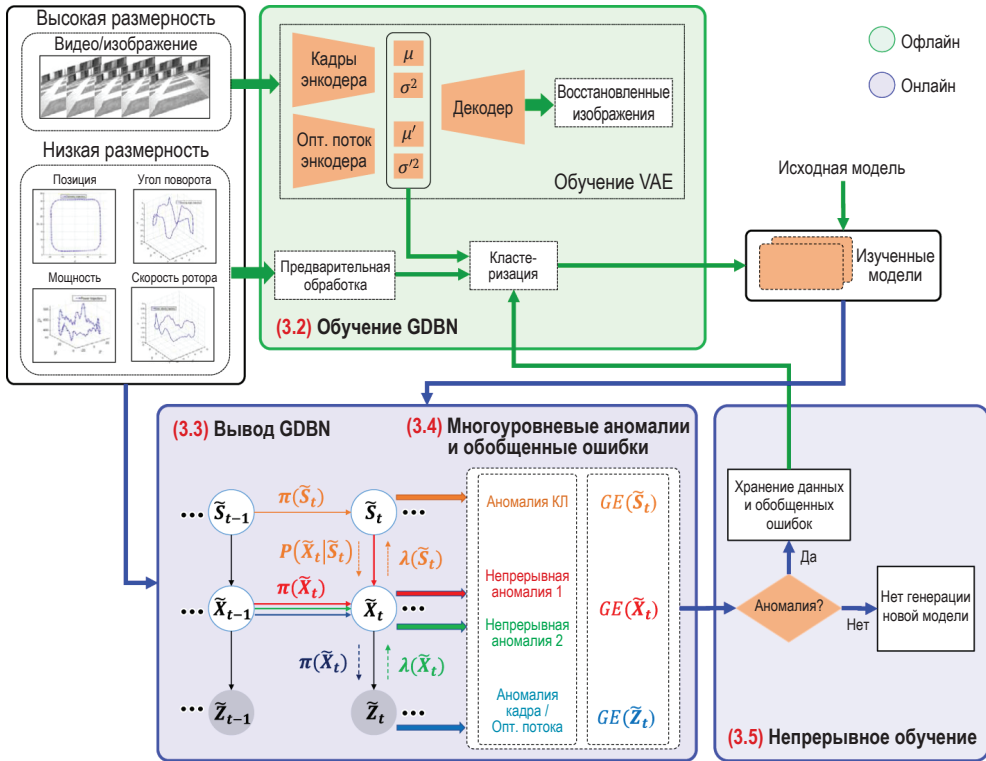


Рис. 13.2 ❖ Блок-схема предложенной нами архитектуры

По рис. 13.2 мы также можем вывести характеристики метода, описанные в табл. 13.1: 1) прежде всего наш метод не нуждается в разметке данных. Вместо этого применяется частичное обучение: аномалия обнаруживается относительно обученной модели; 2) в качестве входных данных могут использоваться данные как низкой, так и высокой размерности. На рис. 13.2 в качестве примеров низкой размерности показаны данные одометрии, угла поворота рулевого колеса, скорости его поворота и мощности транспортного средства; видеоданные с камеры представляют собой данные высокой размерности; 3) рассматриваются данные временного ряда; 4) изучается вероятностная модель в форме GDBN; 5) аномалии относительно этой модели обнаруживаются на разных уровнях абстракции, в частности на уровне кадра, непрерывном и дискретном уровнях; 6) обнаруженные аномалии связаны с доверительным интервалом, как будет показано в разделе 13.3.4; 7) при обнаружении аномалии относительно изученной модели создается новая модель, поддерживающая инкрементное обучение и расширение предыдущих знаний.

В следующих разделах представлено более подробное описание этой архитектуры. Обучение модели GDBN описано в разделе 13.3.2. Разделы 13.3.3 и 13.3.4 описывают алгоритм тестирования MJPF и его многоуровневые измерения аномалий соответственно. Наконец, в разделе 13.3.5 рассказано про

использование обобщенных ошибок, полученных с помощью аномалий DBN, в качестве основы для непрерывного обучения новых моделей. Номер раздела главы, в котором объясняется каждое понятие, также выделен красным цветом на рис. 13.2 рядом с его графическим изображением.

13.3.2. Модель обобщенной динамической байесовской сети (GDBN)

Первоначально самосознающий агент начинает воспринимать окружающую среду с помощью исходной GDBN, т. е. *фильтра с нулевым влиянием* (null force filter) со статическими предположениями о состояниях среды, путем интерпретации полученных обобщенных наблюдений $\tilde{Z}_t = [Z_t \ \dot{Z}_t]$, которые содержат переменную и ее обобщенные координаты движения, поступающие от датчиков агента. В этом случае агент постоянно обнаруживает аномалии и вычисляет обобщенные ошибки $GE(\tilde{X}_t)$ в виде:

$$\tilde{X}_t = [X_t \ \dot{X}_t], \quad (13.5)$$

где X_t – предсказанные скрытые состояния, полученные фильтром с нулевым влиянием в соответствии со следующей динамической моделью:

$$X_t = X_{t-1} + v_t, \quad (13.6)$$

в то время как \dot{X}_t – ошибки в производных, рассчитанные как обновления фильтром с нулевым влиянием с использованием текущих обобщенных наблюдений следующим образом:

$$\dot{X}_t = \frac{Z_t - HX_{t-1}}{\Delta t}. \quad (13.7)$$

Далее GE, собранные в предыдущем опыте, используются на этапе обучения в качестве входных данных для алгоритма кластеризации без учителя (например, Growing Neural Gas (GNG) (Fritzke, 1994), Self-Organizing Maps (SOMs) (Kohonen, 2001)), который кодирует GE в дискретные компоненты, производящие набор дискретных переменных (**S**) или нейронов, которые мы называем *сверхсостояниями* (superstate), так что:

$$\mathbf{S} = \{S_1, S_2, \dots, S_M\}, \quad (13.8)$$

где M – общее количество сверхсостояний.

Заметим, что рассмотренные выше уравнения относятся к случаю данных малой размерности, когда модель наблюдений известна, является линейной и выражается матрицей H , отображающей наблюдения в скрытые состояния. В случае многомерных данных мы должны сначала понизить размерность с помощью экстрактора признаков. Следовательно, первая часть GDBN, которую необходимо изучить, – это сама модель наблюдений. Как говорилось в разделе 13.2.6, это может быть выполнено путем обучения таких сетей,

как VAE и GAN. В данной главе мы рассмотрим в качестве примера использование VAE. Чтобы получить как информацию о содержимом кадра, так и о движении между последовательными кадрами, мы обучаем пару VAE, одну для восстановления кадра в моменты времени t (обозначим ее z_t), а другую для восстановления *оптического потока* (optical flow, OF) между кадрами в моменты времени t и $t + 1$ (OF_t). Соответственно, узкое место архитектуры разделено на две части: одну мы называем X_t – она представляет состояние/контент (от z_t); другую называем \tilde{X}_t – она фиксирует информацию о движении/скорости (из OF_t). Кластеризация выполняется по комбинации этих двух переменных.

После выполнения процесса уменьшения размерности оставшая часть метода остается в основном такой же, с некоторыми поправками и ограничениями.

После кластеризации матрица перехода (П) размерностью $M \times M$, определяемая как

$$P = \begin{bmatrix} \pi(S_t = S_1) \\ \vdots \\ \pi(S_t = S_M) \end{bmatrix} = \begin{bmatrix} \pi_{11} & \cdots & \pi_{1M} \\ \vdots & \ddots & \vdots \\ \pi_{M1} & \cdots & \pi_{MM} \end{bmatrix}, \quad (13.9)$$

изучается путем оценки вероятностей перехода $\pi_{ij} = P(S_t = i | S_{t-1} = j)$, $i, j \in \mathbf{S}$ за период времени. Таким образом, можно создать набор обобщенных сверхсостояний \tilde{S}_t , определенных как

$$\tilde{S}_t = [S_t \ \dot{S}_t] = [S_t \ E(S_t | S_{t-1})] \quad (13.10)$$

и включающих в себя текущую дискретную переменную S_t и событие $E(\cdot)$ перехода к этой переменной, обусловленное нахождением в S_{t-1} в предыдущий момент времени. Этот набор дискретных переменных образует верхний, или дискретный, уровень GDBN. Каждое обобщенное сверхсостояние \tilde{S}_t ($\tilde{S}_t \in \mathbf{S}$) связано со статистическими свойствами, такими как среднее значение ($\mu_{\tilde{S}_t}$), ковариационная матрица ($\Sigma_{\tilde{S}_t}$) и набор скрытых обобщенных состояний (GS) \tilde{X}_t , закодированных внутри него. Скрытые непрерывные обобщенные состояния \tilde{X}_t представляют промежуточный или непрерывный уровень GDBN. Отношение между скрытыми состояниями и сверхсостояниями характеризуется связью $P(\tilde{X}_t | \tilde{S}_t)$ в GDBN. В свою очередь, GDBN представляет скрытые переменные в обобщенных координатах движения, что позволяет самосознающему агенту самоорганизовываться путем оптимизации совместного апостериорного распределения по мере поступления новых наблюдений и непрерывно кодировать новые понятия, связанные с возникающей ситуацией после обнаружения аномалий.

Нижний уровень GDBN соответствует реальным обобщенным наблюдениям \tilde{Z}_t , измеренным датчиками. Путь от \tilde{S}_t к \tilde{Z}_t образует цепь причинно-следственных связей между случайными величинами на иерархических уровнях. Вероятностные зависимости между переменными, участвующими в цепи причинно-следственных связей, характеризуются связанными ребрами.

Учитывая цепное правило, совместную вероятность \tilde{S}_t, \tilde{X}_t и $\tilde{Z}_t, P(\tilde{S}_t, \tilde{X}_t, \tilde{Z}_t)$ можно разложить на множители как произведение условных вероятностей, таких как:

$$P(\tilde{S}_t, \tilde{X}_t, \tilde{Z}_t) = P(\tilde{S}_t)P(\tilde{X}_t|\tilde{S}_t)P(\tilde{Z}_t|\tilde{X}_t). \quad (13.11)$$

Это уравнение подразумевает возможность использования модели для создания новых выборок данных с учетом прямой причины (т. е. родительского узла). Кроме того, связи между обобщенными скрытыми переменными в последовательные моменты времени представляют собой соответствующие условные временные вероятности. Π включает в себя динамические правила на дискретном уровне, которые управляют динамическими изменениями путем переключения между несколькими динамическими моделями на непрерывном уровне. Такое вероятностное причинно-следственное обоснование в GDBN позволяет объяснять события, диагностировать причины и делать прогнозы будущих событий, которые улучшают процессы принятия решений/действий. Вероятность перехода $P(\tilde{S}_t|\tilde{S}_{t-1})$ на дискретном уровне можно разложить следующим образом:

$$\tilde{S}_t = f(\tilde{S}_{t-1}) + w_t = f(\pi_{ij}) + w_t, \quad (13.12)$$

где $f(\cdot)$ – нелинейная функция, определяющая временную эволюцию сверхсостояний на основе изученной матрицы переходов и подверженная шуму процесса w_t , который, как предполагается, получен из нулевого многомерного нормального распределения с ковариантой Σ_t , такой что $w_t \sim \mathcal{N}(0, \Sigma_t)$.

Динамические причинно-следственные модели, описывающие представление в пространстве состояний процесса временных рядов – предполагая, что каждое наблюдение \tilde{Z}_t генерируется из d -мерного скрытого состояния \tilde{X}_t , которое, кстати, было порождено дискретным скрытым сверхсостоянием \tilde{S}_t , – имеют следующий вид:

$$\tilde{X}_t = g(\tilde{X}_{t-1}, \tilde{S}_{t-1}) + w_t = A\tilde{X}_{t-1} + B\tilde{S}_{t-1} + w_t; \quad (13.13)$$

$$\tilde{Z}_t = h(\tilde{X}_t) + v_t = H\tilde{X}_t + v_t. \quad (13.14)$$

Здесь предполагается, что непрерывная функция $g(\cdot)$ в уравнении (13.13) является линейной и определяет эволюцию состояния во времени на непрерывном уровне, руководствуясь предсказаниями дискретного уровня, и подвергается влиянию гауссова шума w_t . В уравнении (13.13) A и B – матрица динамической модели и матрица модели управления соответственно. С каждым сверхсостоянием \tilde{S}_t связан другой управляющий вектор $U_{\tilde{S}_t}$, который зависит от средней производной выборок \tilde{X}_t , закодированных в этом сверхсостоянии. Таким образом, Π кодирует не только переходы между сверхсостояниями, но и переходы между различными линейными моделями на непрерывном уровне. В уравнении (13.14) H – матрица наблюдения, которая отображает скрытое \tilde{X}_t в наблюдение \tilde{Z}_t , а v_t – гауссов шум измерений с нулевым средним и ковариацией R_t , такой что $v_t \sim \mathcal{N}(0, R_t)$.

В случае многомерных данных модель наблюдения, описанная в уравнении (13.14), связана с нелинейным преобразованием, применяемым VAE при

извлечении признаков. Следовательно, функция h в этом случае нелинейна. Кроме того, уравнение (13.13) невозможно непосредственно применить для описания эволюции состояния во времени из-за сильной нелинейности, которая может присутствовать в GS. Чтобы сохранить структуру многомерного случая как можно более похожей на маломерную, для каждого найденного сверхсостояния в нашей архитектуре выполняется изучение нейронной сети $N^{(S)}$, описывающей временную эволюцию GS для этого сверхсостояния. Поэтому уравнение (13.13) можно заменить следующим выражением:

$$\tilde{X}_t = g(\tilde{X}_{t-1}, \tilde{S}_{t-1}) + w_t = N^{\tilde{S}_t}(\tilde{X}_{t-1}) + w_t, \quad (13.15)$$

где w_t – ошибка после сходимости сети, которая может быть аппроксимирована гауссовым шумом.

На рис. 13.3 в обобщенном виде изображена предлагаемая GDBN. Красным показаны внутрикадровые соединения, связанные с соответствующим им элементом: $P(\tilde{Z}_{t-1}|\tilde{X}_{t-1})$ – модель наблюдения, соответствующая матрице H в маломерном случае и декодеру $P_\phi(\tilde{Z}|\tilde{X})$ VAE в многомерном случае; связь $P(\tilde{X}_{t-1}|\tilde{S}_{t-1})$ коррелирует с кластерной ковариантой $\Sigma_{\tilde{S}_t}$. Зеленым цветом показаны межкадровые соединения: прогнозная модель $P(\tilde{X}_t|\tilde{X}_{t-1})$ выбирается через локальное движение $U_{\tilde{S}_t}$ в маломерном случае и через $N_{\tilde{S}_t}$ в многомерном; $P(\tilde{S}_t|\tilde{S}_{t-1})$ кодируется матрицей перехода Π .

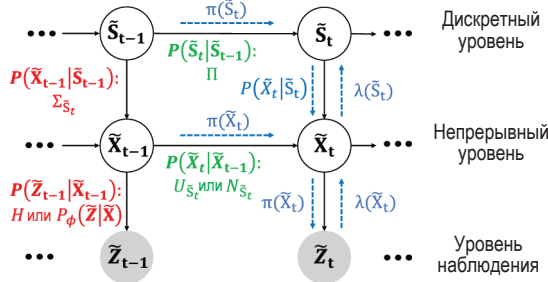


Рис. 13.3 ❖ Предлагаемая обобщенная динамическая байесовская сеть

13.3.3. Алгоритм логического вывода в реальном времени

Марковский скачкообразный фильтр частиц (MJPF), впервые представленный в работе (Baydoun et al., 2018), здесь используется в процессе реального времени для выполнения выводов на разных иерархических уровнях, начиная с изученной модели GDBN. Алгоритм MJPF реализует *коммутирующую модель* (switching model), которая использует комбинацию фильтра частиц (PF) для прогнозирования дискретных сверхсостояний и банка фильтров Калмана (KF) для прогнозирования и оценки непрерывных состояний. В случае GS, извлеченных из многомерных данных, в которых мы применяем нелинейную

модель для прогнозирования состояния, вместо этого можно использовать *сигма-точечный банк фильтров Калмана* (unscented Kalman filters, UKF) (Wan, van der Merwe, 2000), как это сделано в (Slavic et al., 2020). Такая коммутация между динамическими переходами на дискретных/непрерывных уровнях и наблюдениями позволяет обновлять доверие в скрытых переменных, передавая локальные сообщения в режимах одновременного вывода, а именно прогнозирующего или причинно-следственного вывода (сверху вниз) и диагностического вывода (снизу вверх). Временные прогностические сообщения $\pi(\tilde{S}_t)$, $\pi(\tilde{X}_t)$ (рис. 13.3) зависят от динамических правил, хранящихся в модели, в то время как иерархические внутрисрезовые нисходящие сообщения от \tilde{S}_t к \tilde{X}_t зависят от статистики кластеризации, включая средние значения и ковариационные матрицы. Причинность «снизу вверх» основана на моделях правдоподобия, состоящих из сообщений $\lambda(\tilde{S}_t)$, $\lambda(\tilde{X}_t)$ (рис. 13.3), передаваемых в качестве обратной связи для корректировки ожиданий с учетом последовательности наблюдений. В основе фильтра частиц лежит матрица перехода, закодированная в динамической модели как прогнозное распределение для предсказания будущих сверхсостояний (\tilde{S}_t^n). Первоначально он берет N выборок из этого распределения, которые имеют одинаковый вес и связаны с конкретным сверхсостоянием, таким образом, что

$$\langle \tilde{S}_t^n, W^n \rangle \sim \langle \pi(\tilde{S}_t^n), 1/N \rangle. \quad (13.16)$$

Затем для каждой частицы (n) применяется фильтр Калмана для предсказания непрерывных переменных \tilde{X}_t , которые зависят от ожидаемого сверхсостояния, как указано в уравнении (13.13), и могут быть выражены через условную вероятность $P(\tilde{X}_t | \tilde{X}_{t-1}, \tilde{S}_t^n)$. Апостериорная вероятность, связанная с прогнозируемым состоянием, определяется как

$$\pi(\tilde{X}_t) = P(\tilde{X}_t, \tilde{S}_t | \tilde{Z}_{t-1}) = \int P(\tilde{X}_t | \tilde{X}_{t-1}, \tilde{S}_t) \overbrace{\lambda(\tilde{X}_{t-1})}^{P(\tilde{Z}_{t-1} | \tilde{X}_{t-1})} d\tilde{X}_{t-1}. \quad (13.17)$$

Соответственно, как только обнаруживается новое свидетельство \tilde{Z}_t , MJPF может использовать сообщение, распространяющееся в обратном направлении от нижнего уровня к более высоким уровням, для оценки апостериорной вероятности $P(\tilde{X}_t, \tilde{S}_t^n | Z_t)$ следующим образом:

$$P(\tilde{X}_t, \tilde{S}_t | \tilde{Z}_t) = \pi(\tilde{X}_t) \lambda(\tilde{X}_t). \quad (13.18)$$

Следовательно, веса соответствующей частицы могут быть обновлены в соответствии с выражением

$$W_t^n = W_t^n \lambda(\tilde{S}_t), \quad (13.19)$$

а затем нормализованы в соответствии с методом *последовательной повторной выборки коэффициентов* (sequential importance resampling, SIR). В уравнении (13.19) $\lambda(\tilde{S}_t)$ – диагностическое сообщение, распространяемое снизу вверх по иерархии для обновления показателя доверия в скрытых переменных на этом уровне, которое может быть получено следующим образом:

$$\lambda(\tilde{S}_t) = \lambda(\tilde{X}_t)P(\tilde{X}_t, \tilde{S}_t) = P(\tilde{Z}_t|\tilde{X}_t)P(\tilde{X}_t|\tilde{S}_t), \quad (13.20)$$

где $P(\tilde{X}_t|\tilde{S}_t) \sim \mathcal{N}(\mu_{\tilde{S}_k}, \Sigma_{\tilde{S}_k})$ обозначает гауссово распределение со средним значением $\mu_{\tilde{S}_k}$ и ковариацией $\Sigma_{\tilde{S}_k}$. В свою очередь, $\lambda(\tilde{X}_t) \sim \mathcal{N}(\mu_{\tilde{Z}_t}, R)$ обозначает гауссово распределение со средним $\mu_{\tilde{Z}_t}$ и ковариацией R . Произведение $\lambda(\tilde{X}_t)$ и $P(\tilde{X}_t|\tilde{S}_t)$ можно найти с помощью расстояния Бхаттачарии (D_B) следующим образом:

$$D_B(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)) = -\ln \int \sqrt{\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)} d\tilde{X}_t, \quad (13.21)$$

где $\tilde{S}_k \in \tilde{\mathbf{S}}$. Вектор D_λ , содержащий все значения D_B между $\lambda(\tilde{X}_t)$ и всеми сверх-состояниями в множестве $\tilde{\mathbf{S}}$, здесь вычисляется как

$$D_\lambda = [D_B(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_k)), \dots, D_B(\lambda(\tilde{X}_t), P(\tilde{X}_t|\tilde{S}_t = \tilde{S}_L))]. \quad (13.22)$$

Следовательно, вектор $\lambda(\tilde{S}_t)$ с точки зрения вероятности равен

$$\lambda(\tilde{S}_t) = \left[\frac{1/D_\lambda(1)}{1/\sum_{l=1}^L D_\lambda(l)}, \dots, \frac{1/D_\lambda(L)}{1/\sum_{l=1}^L D_\lambda(l)} \right]. \quad (13.23)$$

В классической байесовской фильтрации для нахождения совместного апостериорного распределения на разных иерархических уровнях используются прогностические и диагностические причинные связи. Однако в этом процессе отсутствует необходимый шаг для оценки различий между двумя сообщениями, поступающими в данный узел, на основе вероятностной метрики, которая оценивает *неожиданность*, вызванную наблюдениями, не объяснимыми с точки зрения модели. В следующем разделе показано, как обеспечить самосознающему агенту возможность обнаруживать мультимодальные аномалии, которые являются основой для постоянного обновления знаний о воспринимаемой среде и кодирования новых концепций, связанных с возникающим аномальным поведением.

13.3.4. Измерения мультимодальных аномалий

В предлагаемом подходе классический байесовский вывод (т. е. MJPF) дополнен расширенными функциями, которые используют вероятностное расстояние между нисходящими и восходящими сообщениями, поступающими в общий узел на разных уровнях внутри GDBN, для определения иерархических измерений аномалий. Такие вероятностные расстояния количественно определяют сходство между двумя распределениями вероятностей $\pi(\mathcal{X})$ и $\lambda(\mathcal{X})$ в области \mathcal{X} , которые могут быть непрерывными или дискретными. К наиболее важным вероятностным расстояниям относятся следующие:

- **расстояние Бхаттачария** (D_B) (Bhattacharyya, 1946): это расстояние было предложено для отражения степени несходства между двумя распределениями вероятностей. Оно основано на коэффициенте Бхаттачария BC , который аппроксимирует степень перекрытия между двумя

распределениями. В случае дискретных вероятностных распределений BC определяется следующим образом:

$$BC(\pi(X)\lambda(X)) = \sum_{x \in X} \sqrt{\pi(x)\lambda(x)}, \quad (13.24)$$

а в случае непрерывных распределений:

$$BC(\pi(X)\lambda(X)) = \int \sqrt{\pi(x)\lambda(x)} dx. \quad (13.25)$$

Таким образом, расстояние Бхаттачария D_B определяется через BC следующим образом:

$$D_B = -\ln[BC(\pi(X), \lambda(X))]. \quad (13.26)$$

В любом случае (дискретное и непрерывное распределения) $0 \leq BC \leq 1$ и $0 \leq D_B \leq \infty$;

- **расстояние Хеллингера** (D_H) (Beran, 1977): это расстояние также связано с коэффициентом Бхаттачария BC , может быть найдено как

$$D_H = \sqrt{1 - BC(\pi(x), \lambda(x))} \quad (13.27)$$

и удовлетворяет следующему свойству: $0 \leq D_H \leq \infty$;

- **дивергенция Кульбака–Лейблера** (D_{KL}) (Kullback, Leibler, 1951): D_{KL} – это способ измерения совпадения между двумя распределениями вероятностей. Если два распределения полностью совпадают, то $D_{KL} = 0$, в противном случае оно находится в диапазоне от 0 до ∞ . D_{KL} между двумя дискретными распределениями вероятностей можно найти следующим образом:

$$D_{KL}(\pi(x) \parallel \lambda(x)) = \sum_{x \in X} \pi(x) \log \frac{\pi(x)}{\lambda(x)}, \quad (13.28)$$

в то время как для непрерывных распределений оно определяется равенством

$$D_{KL}(\pi(x) \parallel \lambda(x)) = \int \pi(x) \log \frac{\pi(x)}{\lambda(x)} dx; \quad (13.29)$$

- **дивергенция Брегмана** (B_D) (Bregman, 1967): B_D измеряет расстояние между двумя распределениями, определенными в терминах строго выпуклой функции (ϕ).

$$B_D = \phi(\pi(X)) - \phi(\lambda(X)) - \nabla \phi(\lambda(X))(\lambda(X) - \pi(X)), \quad (13.30)$$

где $\nabla \phi(\lambda(X))$ – градиент ϕ при $\lambda(X)$ и $B_D \geq 0$.

Далее мы используем некоторые из вышеупомянутых вероятностных расстояний в качестве примеров, чтобы связать показатели аномалий с различными уровнями GDBN.

13.3.4.1. Дискретный уровень

Индикатор аномалии на дискретном уровне определяется как расстояние между прогностическим $\pi(\tilde{S}_t)$ и диагностическим $\lambda(\tilde{S}_t)$ сообщениями, поступающими в узел \tilde{S}_t . Такое различие обеспечивает сигнал осведомленности (сознания), указывающий на то, как реальные сенсорные сигналы, полученные из окружающей среды, соотносятся с правилами, закодированными в обобщенной модели. В качестве примера для измерения сходства между двумя дискретными распределениями вероятностей ($\pi(\tilde{S}_t)$ и $\lambda(\tilde{S}_t)$) мы используем симметричную дивергенцию Кульбака–Лейблера. Аномалия *расхождения Кульбака–Лейблера* (KLDA) согласно определению в (Krayani et al., 2020) имеет следующий вид:

$$KLDA = D_{KL}(\pi(\tilde{S}_t) \parallel \lambda(\tilde{S}_t)) + D_{KL}(\lambda(\tilde{S}_t) \parallel \pi(\tilde{S}_t)). \quad (13.31)$$

В каждый момент времени t извлекается гистограмма предсказанных частиц, и вероятность появления каждой частицы рассчитывается как:

$$p(\tilde{S}_t = i) = \frac{y(\tilde{S}_t = i)}{N} \quad i \in \tilde{\mathcal{S}}, \quad (13.32)$$

где $y(\cdot)$ – частота появления определенного сверхсостояния i , а N – общее количество частиц, проходящих через PF. Стоит отметить, что $\lambda(\tilde{S}_t)$ уникально для всех частиц в момент времени t . Определим \mathcal{S} как множество выигрышных частиц, вероятность появления которых больше нуля, такое что:

$$\mathcal{S} = \{i | p(\tilde{S}_t = i) > 0 \quad i \in \tilde{\mathcal{S}}. \quad (13.33)$$

D_{KL} вычисляется между $\lambda(S_t)$ и конкретными строками матрицы перехода, относящимися к частицам-победителям в \mathcal{S} . Следовательно, (13.31) принимает вид:

$$KLDA = \sum_{i \in \mathcal{S}} \left[p(i) \sum_{j=1}^M \pi_{ij} \log \left(\frac{\pi_{ij}}{\lambda_j} \right) \right] + \sum_{i \in \mathcal{S}} \left[p(i) \sum_{j=1}^M \lambda_j \log \left(\frac{\lambda_j}{\pi_{ij}} \right) \right]. \quad (13.34)$$

13.3.4.2. Непрерывный уровень

На этом уровне мы сосредоточимся на сообщениях, поступающих в узел \tilde{X}_t , и рассчитаем соответствующие измерения аномалий, определенные как статистические показатели на основе вероятностного расстояния (например, D_B , D_H , D_{KL} и т. д.). Сообщение $P(\tilde{X}_t | \tilde{S}_t)$, направляемое на промежуточный уровень, описывает вероятность наличия предсказания \tilde{X}_t в определенном сверхсостоянии. Таким образом, вычисление разницы между прогнозирующим сообщением $\pi(\tilde{X}_t)$ и $P(\tilde{X}_t | \tilde{S}_t)$ позволяет оценить, совпадают ли прогнозы, выполненные на дискретном уровне, с прогнозами, сделанными на непрерывном уровне. Например, такую разницу можно получить через расстояние D_B следующим образом:

$$Db2 = D_B(\pi(\tilde{X}_t), P(\tilde{X}_t | \tilde{S}_t^n)) = -\ln \int \sqrt{P(\tilde{X}_t, \tilde{S}_t^n | \tilde{Z}_{t-1}) P(\tilde{X}_t | \tilde{S}_t^n)} d\tilde{X}_t. \quad (13.35)$$

С другой стороны, знание того, насколько реальные наблюдения подтверждают прогнозы, выполняемые на непрерывном уровне, приводит к обнаружению любого аномального поведения, происходящего в окружающей среде. Вторая аномалия на непрерывном уровне может быть рассчитана как разница между наблюдениями $\lambda(\tilde{X}_t)$ и прогнозируемыми обобщенными состояниями $\pi(\tilde{X}_t)$, которая также основана на вероятностных расстояниях, определенных ранее (например, D_B , D_H , D_{KL} и т. д.). Например, используя расстояние D_B , вторую аномалию на непрерывном уровне можно найти как

$$Db1 = D_B(\pi(\tilde{X}_t), \lambda(\tilde{X}_t)) = -\ln \int \sqrt{P(\tilde{X}_t, \tilde{S}_t^n | \tilde{Z}_{t-1}) P(\tilde{Z}_t | \tilde{X}_t)} d\tilde{X}_t. \quad (13.36)$$

13.3.4.3. Уровень наблюдения

Аномалия на этом уровне особенно информативна в случае многомерных данных. В целом можно выделить два типа аномалий на уровне наблюдения: 1) *прямая ошибка реконструкции* из-за изучения модели наблюдения, которая не подходит для наблюдаемых данных. Это, например, случай аномалий из-за незнакомого контента и элементов, появляющихся в сцене. Такие аномалии приводят к большому расхождению между наблюдаемым изображением X_t и его реконструкцией \tilde{X}_t через сеть VAE; 2) расстояние между наблюдением X_t в момент t и прогнозным сообщением $\pi(\tilde{X}_t)$, распространяемым вперед от непрерывного уровня к узлу Z_t . На практике его можно получить путем расчета среднеквадратической ошибки (MSE) между прогнозируемым изображением \tilde{Z}_{t-1} в момент времени $t-1$ и наблюдаемым изображением X_t в момент времени t .

13.3.5. Использование обобщенных ошибок для непрерывного обучения

В предыдущих разделах вы видели, как предложенный подход позволяет дополнить архитектуру GDBN и использовать ее концепцию передачи сообщений для расчета аномалий и обобщенных ошибок (GE).

Индикаторы аномалий позволяют агенту понять, подходят ли используемые им модели для предсказания эволюции мира и его собственного состояния в ситуации, которую он ощущает и переживает. Как показано на рис. 13.2, на графике «непрерывное обучение», при обнаружении высоких аномалий агент получает предупреждение о том, что ему следует изучить новую модель. Следовательно, он хранит соответствующие GE.

С другой стороны, назначением GE является определение того, какие действия фильтр должен выполнить, чтобы скорректировать сделанные им прогнозы. Новая модель, извлеченная из GE, минимизирует ошибки на том же типе последовательности, на котором они были найдены. Процесс кластеризации выполняется заново, и новая модель вставляется в словарь изученных моделей.

Подводя итог, можно сказать, что *аномалии указывают на необходимость создания новой модели*, а обобщенные ошибки инструктируют агента о том, как следует создавать новую модель.

Следует отметить, что в многомерном случае присутствуют некоторые ограничения непрерывного обучения через GE. Могут быть обнаружены новые наблюдения (например, новые объекты, появившиеся на сцене, новые типы окружающей среды и т. д.). В этом случае необходимо обучить новую модель VAE. В подобной ситуации тестирование требует параллельного использования нескольких VAE (и, следовательно, нескольких моделей наблюдения).

13.4. ПРИМЕР: ОБНАРУЖЕНИЕ АНОМАЛИЙ В МУЛЬТИСЕНСОРНЫХ ДАННЫХ ОТ АВТОМОБИЛЯ С САМОСОЗНАНИЕМ

13.4.1. Описание условий эксперимента

В качестве примера, на котором можно протестировать архитектуру, описанную в разделе 13.3, выполняется обнаружение аномалий мультисенсорных данных, получаемых от движущегося автомобиля. Транспортное средство под названием iCab (Marin-Plaza et al., 2016), показанное на рис. 13.4а, управляется человеком при выполнении различных задач в закрытой среде, показанной на рис. 13.4б. Автомобиль служит источником различных данных (например, видео с бортовой передней камеры, данные одометрии, сигналы рулевого управления и т. д.). Мы решили подробно рассмотреть данные одометрии и видеоданные и предоставить дополнительные примеры, касающиеся управляющих данных (угол поворота руля, скорость вращения рулевого вала, мощность). Поскольку одометрические наблюдения имеют размерность $d = 2$, они представляют собой пример сенсорных данных малой размерности. С другой стороны, данные изображения имеют начальную размерность $d = 640 \times 480$ и послужат учебным примером обработки данных высокой размерности. Напомним, что оба этих типа сенсоров мы определили во введении к главе как *экстероцептивные сенсоры*.

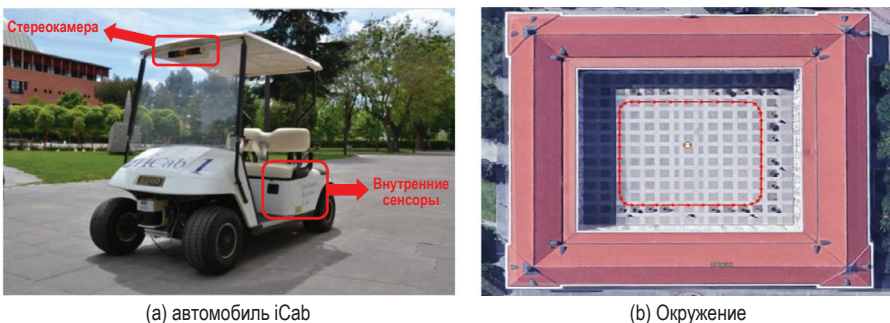


Рис. 13.4 ❖ Используемый в эксперименте автомобиль и окружающая среда

Для проведения обучения на исходной модели, как показано на рис. 13.2, мы берем данные из случая, когда транспортное средство перемещается по окружающей среде без влияния внешних агентов, т. е. осуществляет *следование по периметру* (РМ) двора. Предусмотрены две другие задачи, в которых транспортному средству в его первоначальном мониторинге периметра мешает присутствие пешеходов. Следовательно, поскольку водитель должен выполнять новые движения, чтобы избежать наезда на пешеходов, в потоке данных должны быть обнаружены аномалии относительно исходной модели. Эти две аномальные задачи различаются в зависимости от того, где находится пешеход, и, как следствие, от того, как водитель их избегает: в одном случае (рис. 13.5e) пешехода, находящегося в середине дорожки, избегают путем объезда сбоку (РА); в другом случае (рис. 13.5f), поскольку пешеход находится в углу траектории, его можно избежать за счет U-образного разворота.

На рис. 13.5 изображены упомянутые задачи, включая положения, которые транспортное средство занимает с течением времени (рис. 13.5a, 13.5c, 13.5e слева), и примеры кадров с фронтальной камеры автомобиля (рис. 13.5b, 13.5d, 13.5f справа).

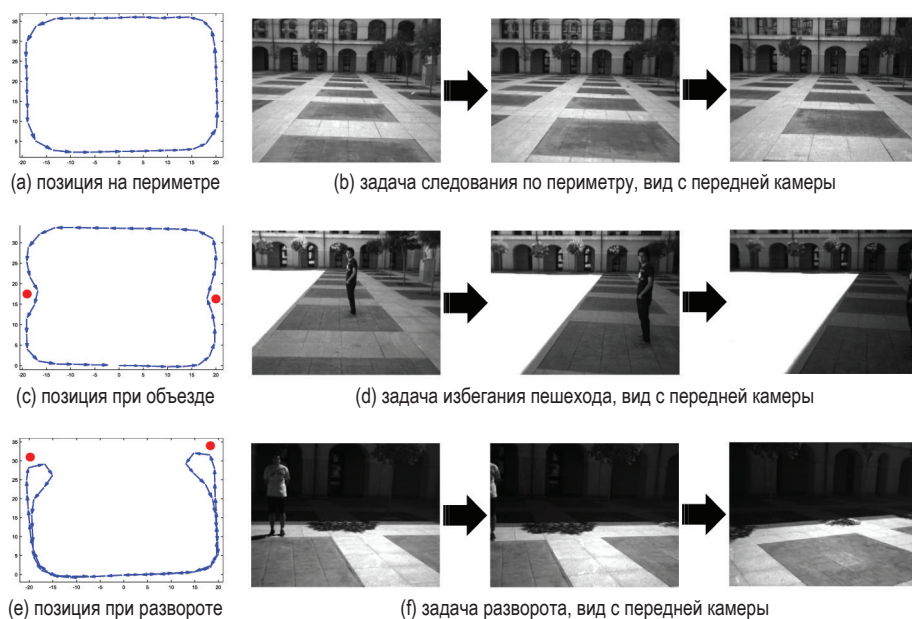


Рис. 13.5 ❖ Тестовые задачи, выполняемые самосознающим транспортным средством. Слева: данные одометра транспортного средства изображены синим цветом, а красные точки показывают, где находится пешеход. Справа: несколько изображений с передней камеры автомобиля при выполнении трех задач

13.4.2. Обучение модели DBN

В качестве первого шага в рассматриваемой архитектуре необходимо обучить начальную модель. В случае данных одометрии модель наблюдения извест-

на и задается матрицей H , определенной, как показано в разделе 13.3.2. Компоненты модели, которые необходимо изучить: 1) дискретные сверхсостояния, найденные с помощью алгоритма кластеризации, такого как GNG (Fritzke, 1994), с их статистическими свойствами (т. е. среднее значение μ_{S_k} , ковариационная матрица Σ_{S_k}); 2) матрица перехода Π , описывающая связи между сверхсостояниями. В случае видеоданных необходимо также изучить параметры кодеров и декодеров VAE, что приводит к вычислению модели наблюдения DBN. Кроме того, во втором случае необходимо также обучить нейронные сети, формирующие модель прогнозирования внутри каждого кластера.

На этом этапе изучается модель *нормальности*. В нашем исследовании для получения этой исходной модели использованы данные следования по периметру (PM).

На рис. 13.6 показаны результаты кластеризации по данным одометрии (13.6a) и видео (13.6b). В случае одометрии средние значения μ_{S_k} для каждого кластера можно визуальнo распознать как среднее значение данных о положении кластеров (показаны голубыми точками) и средние значения кластеризованных данных скорости (показаны синими стрелками). Кроме того, пунктирные кружки пропорциональны стандартному отклонению положений кластеров, которое можно извлечь из диагонали Σ_{S_k} . Обратите внимание, что для лучшей наглядности окружность была аппроксимирована через максимальное значение в двух направлениях. В свою очередь, на рис. 13.6a представлена кластеризация данных с видеокамеры. Так как противоположные стороны двора имеют схожий внешний вид и выполняются одни и те же движения, то присутствует симметрия. Теневой области назначается уникальный отдельный кластер (кластер 1).

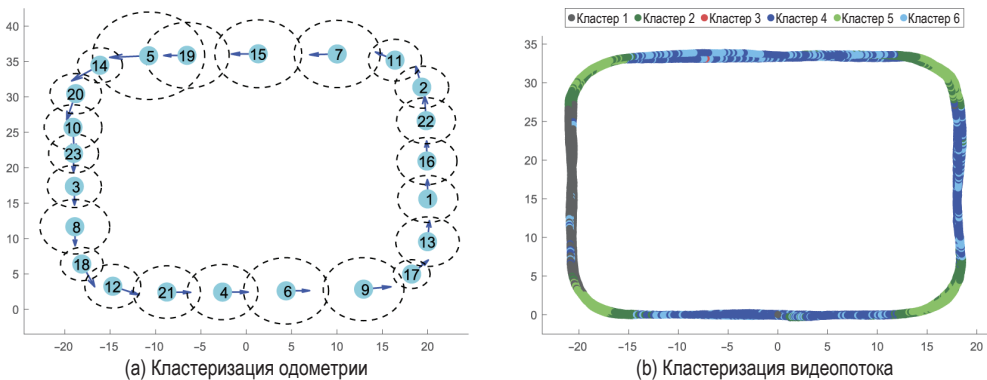


Рис. 13.6 ❖ Одометрия и кластеризация видео

13.4.3. Многоуровневое обнаружение аномалий

После того как все параметры исходной модели изучены, ее можно использовать для вывода, когда имеются данные двух других задач. Аномалии представляют собой расхождения между прогнозами, выполненными с помощью

модели на основе задачи следования по периметру, и наблюдениями из данных, полученных во время задач объезда пешехода и U-образного разворота.

13.4.3.1. Задача объезда пешеходов

На рис. 13.7 и 13.8 показаны аномалии, обнаруженные при фильтрации данных одометрии и видеопотока описанной выше структурой. В первом случае отображаются только аномалии на непрерывном и дискретном уровнях, а во втором также отображаются аномалии на уровне наблюдения. Зеленые области относятся к зонам прямолинейного движения, желтые области – к зонам криволинейного движения, а синие области – к зонам, отображающим ненормальные действия (например, маневр уклонения от пешеходов).

На рис. 13.7 изображены три графика. Черный график отображает среднее значение обобщенной ошибки, которое мы обозначили за $\tilde{\epsilon}_t$. Красный график показывает аномалию $Db2$ (уравнение (13.35)). Бросается в глаза сходство между этими двумя графиками, поскольку они оба представляют собой сравнение между $\pi(X_k)$ и $\lambda(X_k)$. Оба графика отображают высокие значения аномалии, соответствующие объезду пешеходов, и ложные срабатывания из-за небольших отклонений во время движения. Однако следует отметить, что $Db1$ дает лучшие результаты, поскольку учитывает полную информацию о вероятности $\pi(X_k)$ и $\lambda(X_k)$, в то время как $\tilde{\epsilon}_t$ отбрасывает информацию о ковариации Σ_{S_k} и сохраняет только среднее значение μ_{S_k} .

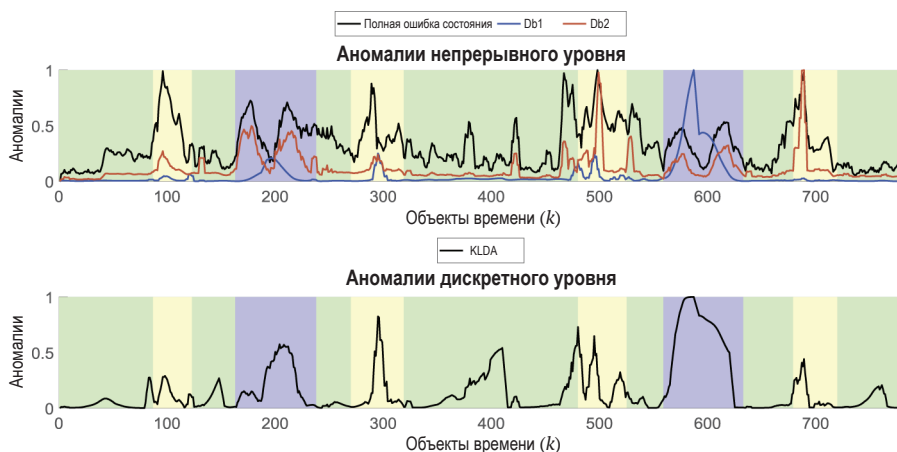


Рис. 13.7 ❖ Значения многоуровневой аномалии одометрии в задаче объезда пешеходов

Наконец, синий график на рис. 13.7 относится к аномалии $Db1$ (уравнение (13.36)). Эта аномалия становится особенно актуальной в центре аномального движения. Например, мы можем наблюдать, что примерно во время $t = 600$ частицы неравномерно распределяются по кластерам относительно кривой в правом верхнем углу рис. 13.6а из-за неоднородности движения в этой области.

Второй график на рис. 13.7 относится к аномалии *KLDA*, описанной уравнением (13.34). Зоны избегания пешеходов явно демонстрируют очень высокое значение аномалии, являющееся следствием необычных проходов между кластерами.

На рис. 13.8 показаны те же типы аномалий в случае с видеоданными. Кроме того, на первом графике показаны аномалии на уровне наблюдения. Черные и синие линии отражают прямые аномалии восстановления VAE, сигнализируя о неподходящей модели наблюдения. Красные и зеленые линии аномалий – это ошибки на уровне изображения из-за несоответствия между наблюдением и прогнозом.

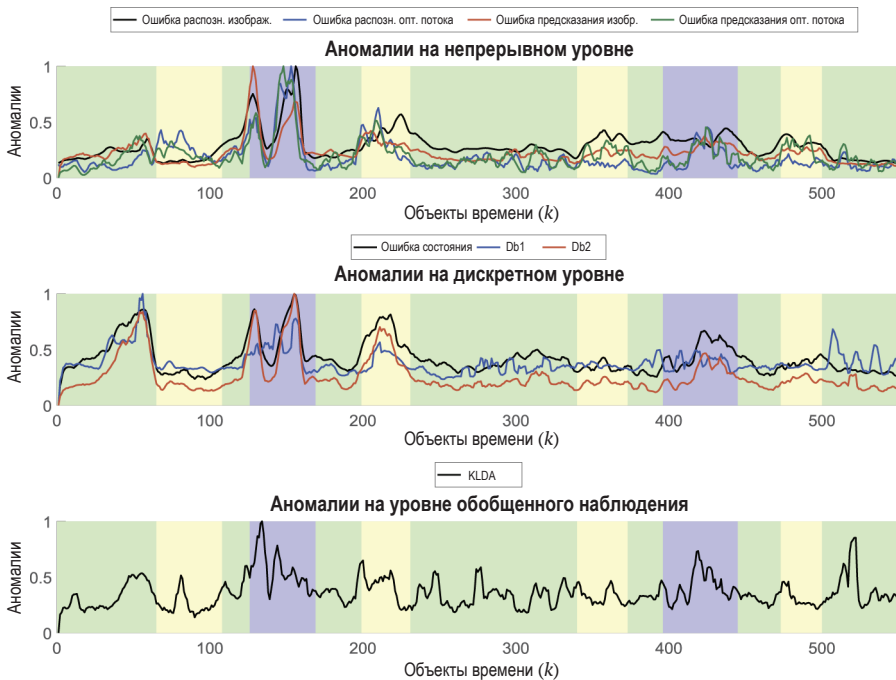


Рис. 13.8 ❖ Значения многоуровневой аномалии видео в задаче объезда пешеходов

13.4.3.2. Задача разворота

На рис. 13.9 и рис. 13.10 показаны аномалии одометрии и видео в задаче U-образного разворота. В этом случае синяя область показывает зону, в которой выполняется разворот. В области между этими двумя зонами автомобиль движется аналогично тому, как он двигался при обучении модели, но в противоположном направлении. Отсюда следует вывод, что: 1) аномалии ϵ_t и *Db2* в случае одометрии (рис. 13.9) имеют высокие значения только в случае аномального разворота и движения в противоположном направлении; 2) аномалия *Db1* и *KLDA* показывают высокое значение на всем протяжении разворота и при движении автомобиля в направлении, противоположном направлению обучения. В обоих случаях это происходит из-за того, что

частицы относятся к кластерам на другой стороне полигона, что приводит к большому расхождению между значениями кластера и прогнозами для CLA и аномальным переходам для *KLDA*; 3) в случае с видео (рис. 13.10) высокие аномалии появляются только в связи с движением разворота и движением по кривым в направлении, противоположном начальному направлению.

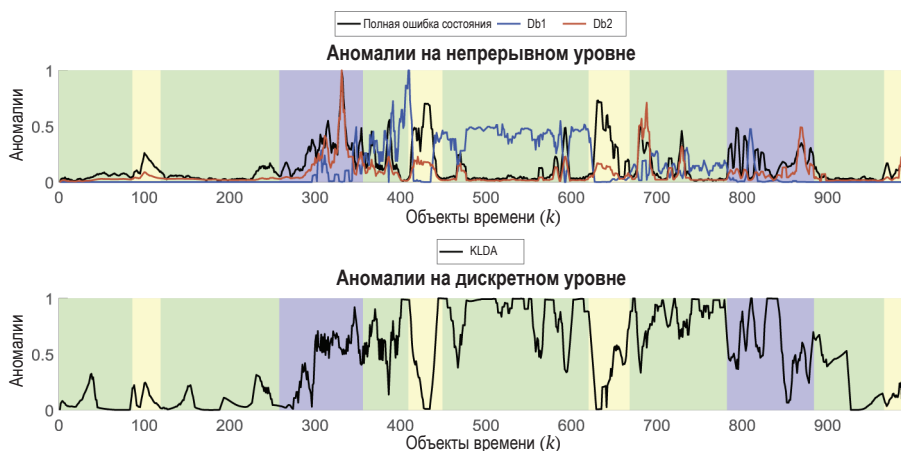


Рис. 13.9 ❖ Значения многоуровневых аномалий одометрии в задаче избегания пешеходов с помощью разворота

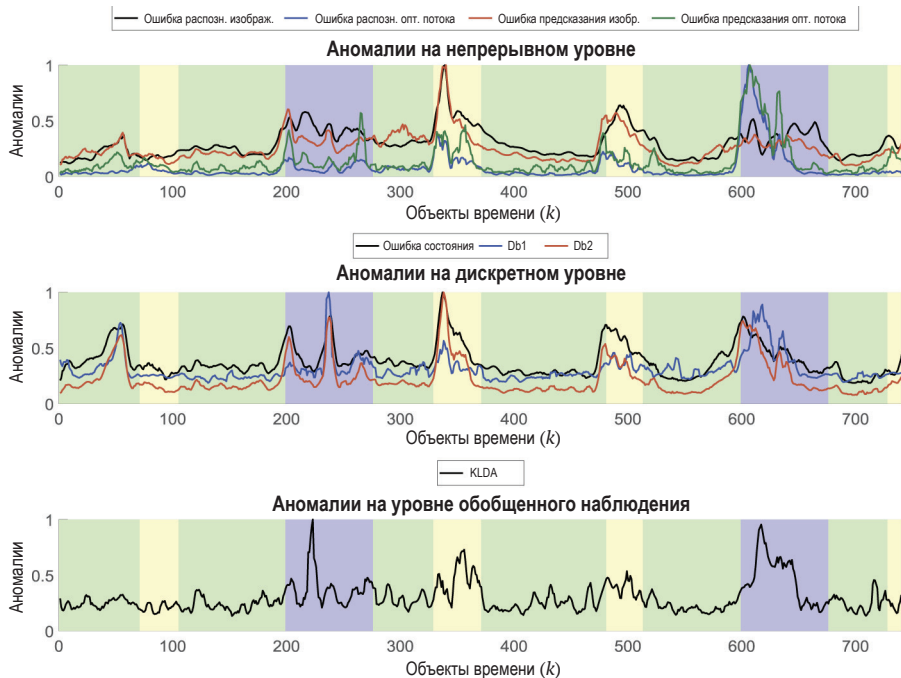


Рис. 13.10 ❖ Значения многоуровневой аномалии видео в задаче избегания пешеходов с помощью разворота

13.4.3.3. Аномалии на уровне изображения

На рис. 13.11 показаны примеры поведения модели на уровне изображения. В двух первых и двух последних столбцах показаны фактические кадры и данные оптического потока в последовательные моменты времени; в третьем и четвертом столбцах показаны результаты прямого восстановления через VAE; в пятом и шестом столбцах показано восстановление предсказанного изображения.

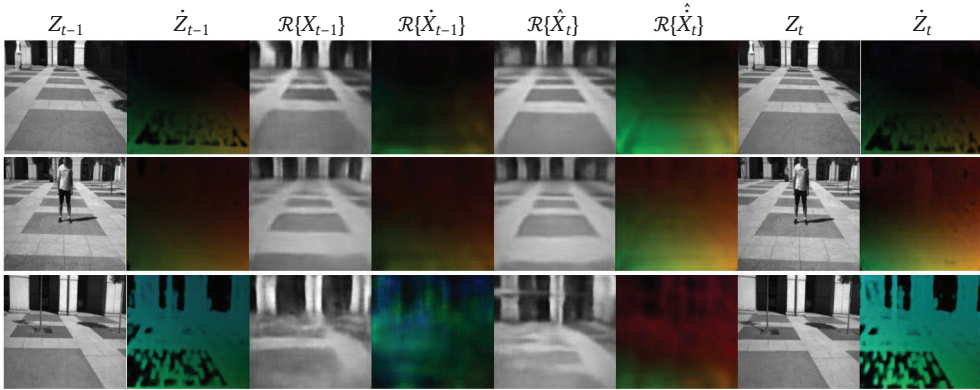


Рис. 13.11 ❖ Примеры аномалий на уровне изображения. С помощью записи $\mathcal{R}\{X\}$ кратко обозначено декодирование скрытой переменной X через декодер VAE

На рис. 13.11 показаны три примера. 1. В первой строке приводится пример нормального прямолинейного движения, взятого из набора данных уклонения от пешехода. 2. Во второй строке мы отображаем момент времени до начала маневра уклонения. Присутствие пешехода вызывает аномалию, строго связанную с моделью наблюдения. Новый VAE должен быть обучен понятию «пешеход». 3. В третьем ряду у нас есть аномалия из-за движения по кривой в противоположном направлении по отношению к направлению при обучении на наборе данных разворота. Нужно заметить, что, несмотря на то что восстановление не является оптимальным, оно является разумным. С другой стороны, прогнозная модель может фиксировать аномалию более очевидным образом. Это связано с тем, что при наблюдении за частью двора, относящейся к траектории, мы могли бы ожидать, что будем двигаться влево (см. красный цвет прогнозируемого оптического потока $\mathcal{R}\{\dot{\hat{X}}_t\}$), но вместо этого мы выполняем поворот вправо.

13.4.3.4. Оценка обнаружения аномалий

Чтобы оценить метод обнаружения аномалий, значение аномалии, полученное на разных уровнях абстракции, сравнивается с эталонным (ground truth, GT). Эталоны сценариев уклонения от пешехода и U-образного разворота строятся вручную с учетом движения автомобиля по данным одометрии.

В случае одометрии U-образного разворота вся последовательность от начала первого разворота до конца второго считается аномальной, так как транспортное средство движется в противоположном направлении относительно изученного. В видеоданных 10 кадров до начала аномального маневра добавляются к положительному эталону, чтобы отметить присутствие пешехода; кроме того, для учета аномальных теневых зон используется простой метод обнаружения теней.

На рис. 13.12 и 13.13 показаны кривые рабочих характеристик приемника (receiver operating characteristic, ROC) для результатов обнаружения аномалий в одометрии и видеопотоке соответственно. Кривая ROC отображает значения частоты истинных положительных результатов (TPR) и частоты ложных срабатываний (FPR) при различных пороговых значениях аномалии, где $TPR = \frac{TP}{TP + FN}$ и $FPR = \frac{FP}{FP + TN}$. В нашем случае TPR и FPR представляют собой процент случаев, в которых аномалии были правильно и неправильно идентифицированы соответственно. Показатель AUC (area under the curve, площадь под кривой) можно использовать для измерения точности метода. Еще одним показателем является сходимость (ассурагу, ACC), определяемая суммой истинно положительных результатов (TP) и истинно отрицательных результатов (TN). В области инкрементных систем с самосознанием могут использоваться дополнительные методы оценки, например насколько хороши аномалии для извлечения ошибок, позволяющих строить лучшие модели.

На рис. 13.12 два непрерывных значения аномалии (т. е. *Db1* и *Db2*) нормализованы и суммированы для получения единого значения аномалии, демонстрирующего гораздо более высокую точность, чем два его отдельных компонента.

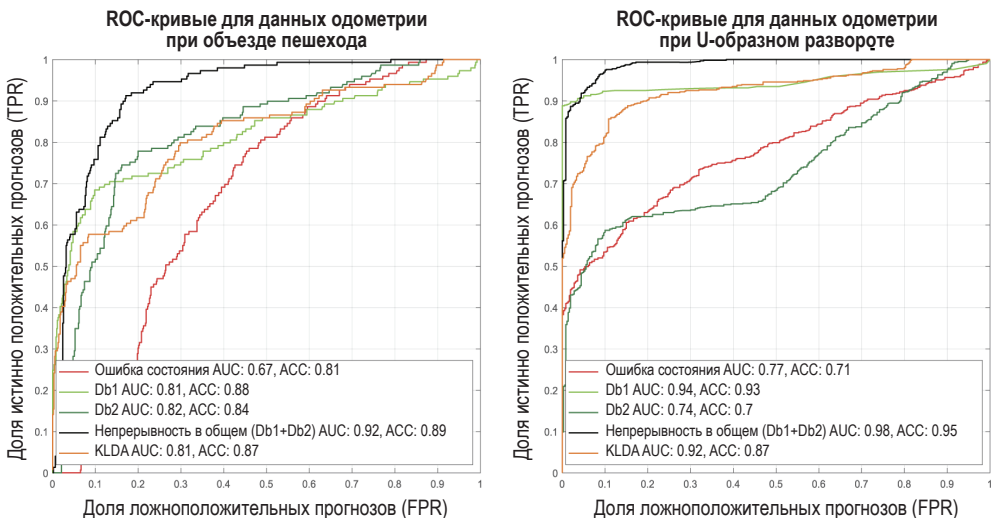


Рис. 13.12 ❖ ROC-кривые для данных одометрии

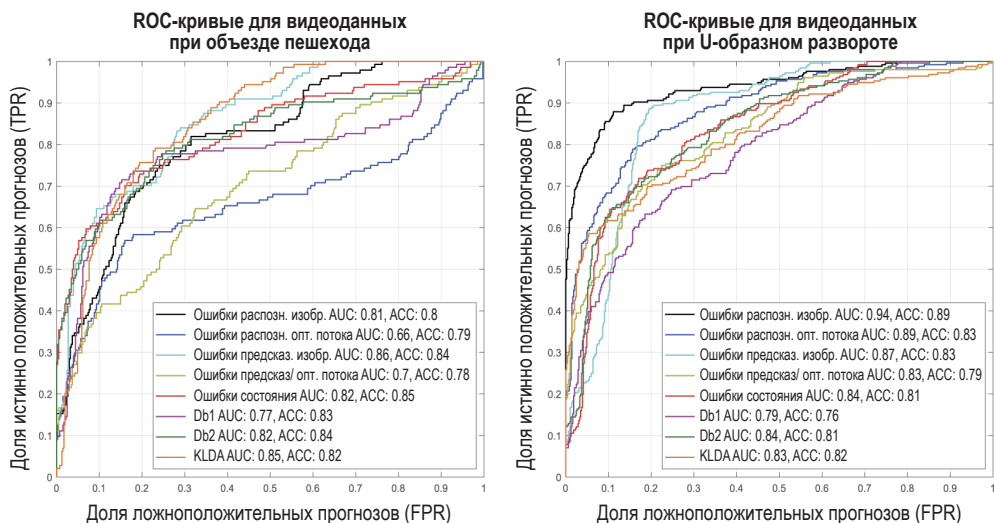


Рис. 13.13 ❖ ROC-кривые для видеоданных

13.4.4. Аномалии проприоцептивных сенсорных данных

Подробные описания примеров обучения и тестирования с использованием одометрии и видеоданных были представлены в предыдущих разделах. Все они были основаны на данных, поступающих с экстероцептивных сенсоров транспортного средства. Помимо экстероцептивных сенсоров, в архитектуре самосознания следует учитывать и проприоцептивные датчики транспортного средства. К ним относятся, например, угол поворота рулевого колеса (S), скорость вращения рулевого вала (V) и мощность (P).

На рис. 13.14 и 13.15 показаны результаты обнаружения аномалии *Db2* для случая уклонения от пешеходов и разворота соответственно. Три контрольных признака, т. е. S, V и P, можно по-разному комбинировать для формирования рассматриваемых сенсорных данных, на основе которых строится модель GDBN. На рис. 13.14 представлены результаты использования комбинаций SP (угол поворота и мощность) и SV (угол поворота и скорость вращения вала) в задаче уклонения от пешеходов; на рис. 13.15 показаны комбинации SP, SV и VP (скорость вращения вала и мощность) на задаче разворота. В обоих случаях зеленые области соответствуют нормальным зонам, а синие – аномальным, т. е. объезду пешеходов в первом случае и развороту во втором. Результаты, относящиеся к управляющим данным¹, можно найти в (Kanapram et al., 2019).

¹ Эти данные называются управляющими, потому что свидетельствуют о работе органов управления (угол и скорость поворота руля, а также нажатия на педаль газа, от которого зависит мощность двигателя). Очевидно, что действия водителя, выраженные в управляющих данных, прямо зависят от аномалий окружающей среды. – Прим. перев.

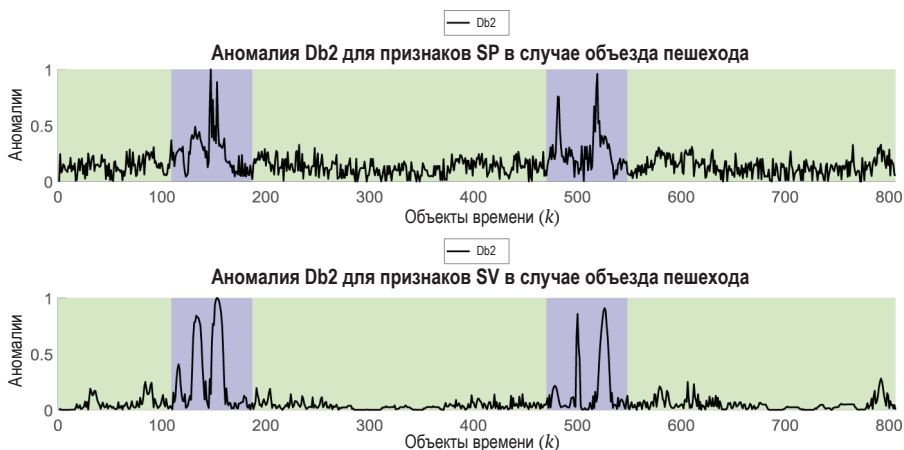


Рис. 13.14 ❖ Аномалия управляющих данных (SP, SV) на непрерывном уровне для случая уклонения от пешеходов

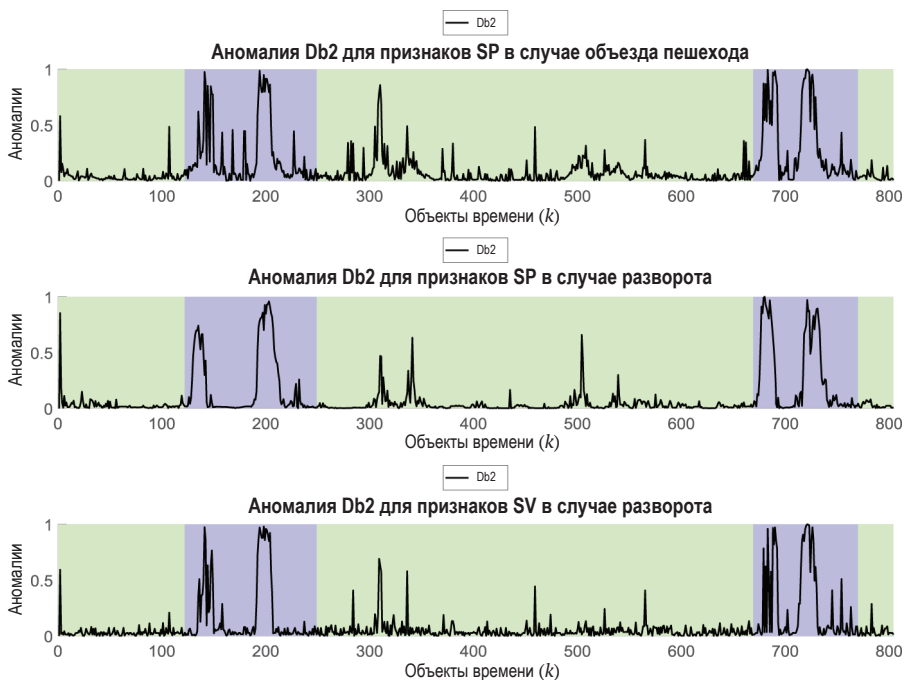


Рис. 13.15 ❖ Аномалия контрольных данных (SP, SV, VP) на непрерывном уровне для случая разворота

13.4.5. Дополнительные результаты

В упомянутом выше примере рассматривались аномалии, которые возникают при движении полуавтономного транспортного средства, вырабаты-

вающего малоразмерные, многомерные, проприоцептивные и экстероцептивные данные. Мы говорили о системе с одним агентом (транспортным средством), но можно было обрабатывать и системы с несколькими агентами, а взаимодействие между ними моделировать на более высоком уровне, как это сделано в недавнем исследовании (Knapram et al., 2020).

Важно отметить, что описанный метод можно применять и к данным из других областей, например к радиооборудованию с самосознанием. Результаты в этом направлении можно изучить в работе (Krayani et al., 2020).

Во всех упомянутых работах рассматривались аномалии максимум на трех уровнях. Однако могут быть определены более высокие уровни абстракции и, следовательно, более высокие уровни аномалий. Одним из примеров является работа (Zaal et al., 2019): для каждой задачи получен граф, соединяющий соседние кластеры; там, где на графе обнаруживается высокая аномалия, возникает новый концепт¹.

13.5. Выводы

В этой главе мы представили архитектуру самосознающей системы обнаружения аномалий в данных временных рядов. В рамках самосознания агент, работающий с мультимедийными данными, включая данные высокой и низкой размерностей, способен обнаруживать аномалии на иерархических уровнях, что считается важным шагом непрерывного обучения новой динамической модели, которая кодирует возникающее аномальное поведение. В случае работы с многомерными данными для уменьшения размерности применяется VAE, которая извлекает признаки и включает их в обобщенный вектор состояния. В случае малоразмерных данных признаки извлекаются непосредственно из наблюдений. Обучение динамической модели в форме вероятностной графовой модели, структурированной в динамической байесовской сети (DBN), образует мост между данными высокой и низкой размерностей. В главе было показано, как можно использовать сообщения, проходящие внутри DBN, для определения иерархии аномалий на основе вероятностных расстояний. Для практической проверки архитектуры был использован реальный набор данных, и результаты показывают, что самосознающий агент способен эффективно обнаруживать мультимодальные аномалии.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Aggarwal C. C., 2016. Outlier Analysis, 2nd ed. Springer Publishing Company, Incorporated.
- Alshazly H., Linse C., Barth E., Martinetz T., 2019. Handcrafted versus cnn features for ear recognition. Symmetry 11, 1493.

¹ В робототехнике и ИИ концепт – это элемент представления знаний. В данном случае он требует дополнительного изучения.

- Antipov G., Berrani S. A., Ruchaud N., Dugelay J. L.*, 2015. Learned vs. hand-crafted features for pedestrian gender recognition. In: ACM International Conference on Multimedia, pp. 1263–1266.
- Baydoun M., Campo D., Sanguineti V., Marcenaro L., Cavallaro A., Regazzoni C.*, 2018. Learning switching models for abnormality detection for autonomous driving. In: 2018 21st International Conference on Information Fusion (FUSION), pp. 2606–2613.
- Becker-Ehmck P., Peters J., Smagt P. V. D.*, 2019. Switching linear dynamics for variational Bayes filtering. In: International Conference on Machine Learning, pp. 553–562.
- Beran R.*, 1977. Minimum Hellinger distance estimates for parametric models. The Annals of Statistics 5, 445–463.
- Bhattacharyya A.*, 1946. On a measure of divergence between two multinomial populations. Sankhyā: The Indian Journal of Statistics (1933–1960) 7, 401–406.
- Bregman L.*, 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. U.S.S.R. Computational Mathematics and Mathematical Physics 7, 200–217.
- Bronstein A., Das J., Duro M., Friedrich R., Kleyner G., Mueller M., Singhal S., Cohen I.*, 2001. Self-aware services: using Bayesian networks for detecting anomalies in Internet-based services. In: 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470), pp. 623–638.
- Campo D., Slavic G., Baydoun M., Marcenaro L., Regazzoni C.*, 2020. Continual learning of predictive models in video sequences via variational autoencoders. In: IEEE International Conference on Image Processing, pp. 753–757.
- Chandola V., Banerjee A., Kumar V.*, 2009. Anomaly detection: a survey. ACM Computing Surveys 41, 15:1–15:58.
- Chong Y. S., Tay Y. H.*, 2015. Modeling representation of videos for anomaly detection using deep learning: a review. arXiv:1505.00523 [abs].
- Dalal N., Triggs B.*, 2005. Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893.
- Damasio A. R.*, 1999. The Feeling of What Happens: Body and Emotion in the Making of Consciousness. Harcourt Brace.
- Foorthuis R.*, 2020. On the nature and types of anomalies: a review. arXiv:2007.15634 [abs].
- Fox E., Sudderth E. B., Jordan M. I., Willsky A. S.*, 2011. Bayesian nonparametric inference of switching dynamic linear models. IEEE Transactions on Signal Processing 59, 1569–1585.
- Fraccaro M., Kamronn S., Paquet U., Winther O.*, 2017. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In: Conference on Neural Information Processing Systems, pp. 3601–3610.
- Friston K. J., Sengupta B., Auletta G.*, 2014. Cognitive dynamics: from attractors to active inference. Proceedings of the IEEE 102, 427–445.
- Fritzke B.*, 1994. A growing neural gas network learns topologies. In: Conference on Neural Information Processing Systems, pp. 625–632.

- Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014. Generative adversarial nets. In: Conference on Neural Information Processing Systems, pp. 2672–2680.
- Haykin S., Fuster J. M., 2014. On cognitive dynamic systems: cognitive neuroscience and engineering learning from each other. *Proceedings of the IEEE* 102, 608–628.
- Johnson M., Duvenaud D., Wiltchko A. B., Adams R., Datta S., 2016. Composing graphical models with neural networks for structured representations and fast inference. In: Conference on Neural Information Processing Systems, pp. 2946–2954.
- Kanapram D., Campo D., Baydoun M., Marcenaro L., Bodanese E., Regazzoni C., Marchese M., 2019. Dynamic Bayesian approach for decision-making in ego-things. In: 2019 IEEE 5th World Forum on Internet of Things (WFIoT), pp. 909–914.
- Kanapram D., Patrone F., Marín-Plaza P., Marchese M., Bodanese E. L., Marcenaro L., Gómez D. M., Regazzoni C. S., 2020. Collective awareness for abnormality detection in connected autonomous vehicles. *IEEE Internet of Things Journal* 7, 3774–3789.
- Kingma D. P., Welling M., 2014. Auto-encoding variational Bayes. In: International Conference on Learning Representations.
- Kingma D. P., Welling M., 2019. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning* 12, 307–392.
- Kiran B. R., Thomas D., Parakkal R., 2018. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging* 4, 36.
- Kohonen T., 2001. Self-Organizing Maps. Ser. Physics and Astronomy Online Library. Springer, Berlin, Heidelberg.
- Krayani A., Baydoun M., Marcenaro L., Alam A. S., Regazzoni C., 2020. Self-learning Bayesian generative models for jammer detection in cognitive-uav-radios. In: GLOBECOM 2020–2020 IEEE Global Communications Conference, pp. 1–7.
- Kullback S., Leibler R. A., 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 22, 79–86.
- Marín-Plaza P., Beltrán J., Hussein A., Musleh B., Martín D., de la Escalera A., Armingol J. M., 2016. Stereo vision-based local occupancy grid map for autonomous navigation in ros. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 703–708.
- Mascaro S., Nicholson A., Korb K., 2014. Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning* 55, 84–98.
- Mihajlovic V., Petkovic M., 2001. Dynamic Bayesian Networks: a State of the Art. TR-CTIT-34 of CTIT Technical Report Series. University of Twente.
- Morin A., 2006. Levels of consciousness and self-awareness: a comparison and integration of various neurocognitive views. *Consciousness Cognition* 15, 358–371.
- Nugroho K. A., 2018. A comparison of handcrafted and deep neural network feature extraction for classifying optical coherence tomography (oct) images. In: International Conference on Informatics and Computational Sciences, pp. 1–6.
- Ramachandra B., Jones M., Vatsavai R. R., 2020. A survey of single-scene video anomaly detection. ArXiv. arXiv: 2004.05993 [abs].

- Ravanbakhsh M., Baydoun M., Campo D., Marín P., Martín D., Marcenaro L., Regazzoni Carlo*, 2020. Learning self-awareness for autonomous vehicles: exploring multisensory incremental models. *IEEE Transactions on Intelligent Transportation Systems*, 1–15.
- Regazzoni C. S., Marcenaro L., Campo D., Rinner B.*, 2020. Multisensorial generative and descriptive self-awareness models for autonomous systems. *Proceedings of the IEEE* 108, 987–1010.
- Rivera A. R., Khan A., Bekkouch I. E. I., Sheikh T. S.*, 2020. Anomaly detection based on zero-shot outlier synthesis and hierarchical feature distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 1–11.
- Salotti J. M.*, 2018. Bayesian network for the prediction of situation awareness errors. *International Journal of Human Factors Modelling and Simulation* 6, 119–126.
- Slavic G., Baydoun M., Campo D., Marcenaro L., Regazzoni C.*, 2021. Multilevel anomaly detection through variational autoencoders and Bayesian models for self-aware embodied agents. *IEEE Transactions on Multimedia*, 1. <https://doi.org/10.1109/TMM.2021.3065232>.
- Slavic G., Campo D., Baydoun M., Marín P., Martín D., Marcenaro L., Regazzoni C.*, 2020. Anomaly detection in video data based on probabilistic latent space models. In: *IEEE Conference on Evolving and Adaptive Intelligent Systems*, pp. 1–8.
- Wan E.A., van der Merwe R.*, 2000. The unscented Kalman filter for nonlinear estimation. In: *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158.
- Wang H., Bah M. J., Hammad M.*, 2019. Progress in outlier detection techniques: a survey. *IEEE Access* 7, 107964–108000.
- Watter M., Springenberg J. T., Boedecker J., Riedmiller M.*, 2015. Embed to control: a locally linear latent dynamics model for control from raw images. In: *Conference on Neural Information Processing Systems*, pp. 2746–2754.
- Winn J., Bishop C.*, 2005. Variational message passing. *Journal of Machine Learning Research* 6, 661–694.
- Zaal H., Iqbal H., Campo D., Marcenaro L., Regazzoni C. S.*, 2019. Incremental learning of abnormalities in autonomous systems. In: *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8.

ОБ АВТОРАХ ГЛАВЫ

Карло Регаццони – профессор когнитивных телекоммуникационных систем в DITEN, Университет Генуи, Италия. Он отвечал за несколько национальных и финансируемых ЕС исследовательских проектов. В настоящее время является координатором международных курсов докторантуры по интерактивным и когнитивным средам с участием нескольких европейских университетов. Был председателем на нескольких конференциях и рецензентом / приглашенным редактором в нескольких международных технических журналах. Занимал множество должностей в руководящих органах IEEE SPS, а в 2015–2017 гг. был вице-президентом Общества обработки сигналов IEEE.

Али Краяни получил степень бакалавра в области телекоммуникаций в Туринском политехническом университете в 2014 г. и степень магистра в области телекоммуникаций в Флорентийском университете в 2017 г. В настоящее время он готовится получить степень доктора философии в рамках Совместной программы докторантуры в области интерактивных и когнитивных сред Университета Генуи и Лондонского университета королевы Марии. Его текущие исследовательские интересы включают когнитивное радио, сотовые системы, связь с БПЛА, самосознание, динамические байесовские сети и искусственный интеллект.

Джулия Славик получила звание инженера по электронике и информационным технологиям в 2017 г. и магистра интернет- и мультимедийных технологий в 2020 г. в Университете Генуи, Италия. В настоящее время является аспиранткой Университета Генуи. Ее исследовательские интересы включают использование алгоритмов глубокого обучения и обработки сигналов для потоковых мультисенсорных данных.

Лучио Марченаро имеет более 20 лет опыта в области обработки сигналов и анализа последовательности изображений. Является автором около 160 научных работ, связанных с обработкой сигналов для компьютерного зрения и когнитивного радио. Окончил факультет электроники в 1999 г. и получил докторскую степень в области компьютерных наук и электронных технологий в 2003 г. Является адъюнкт-профессором факультета телекоммуникаций Политехнической школы Университета Генуи. Его основные текущие исследовательские интересы связаны с обработкой видео для распознавания событий, обнаружением и локализацией объектов в сложных сценах, а также с распределенными гетерогенными сенсорами и когнитивными автономными системами.

Глава 14

Методы PnP и глубокой развертки для восстановления изображения

Авторы главы:
Кай Чжан и Раду Тимофте,
Лаборатория компьютерного зрения,
ETH Zürich, Цюрих, Швейцария

Краткое содержание главы:

- обсуждение достоинств и недостатков моделей и методов на основе глубокого обучения для восстановления изображений;
- методы plug-and-play (PnP) и развертки на основе глубокого обучения могут использовать преимущества как методов, основанных на обучении, так и методов, основанных на моделях;
- экспериментальные результаты демонстрируют гибкость и эффективность методов PnP и развертки для восстановления изображения.

14.1. ВВЕДЕНИЕ

Задача *восстановления изображения* (image restoration, IR) давно привлекает пристальное внимание исследователей из-за ее высокой практической значимости в различных приложениях низкоуровневого зрения (Richardson, 1972; Andrews, Hunt, 1977). В этой главе основное внимание уделяется трем репрезентативным и фундаментальным проблемам IR: *шумоподавлению* (denoising), *удалению размытия* (deblurring) и *сверхразрешению* (superresolution). В общем случае цель IR состоит в том, чтобы восстановить скрытое чистое изображение \mathbf{x} из его ухудшенного наблюдения $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}$, где \otimes пред-

ставляет собой двумерную свертку \mathbf{x} с ядром размытия \mathbf{k} , \downarrow_s обозначает стандартную s -кратную понижающую дискретизацию, т. е. сохранение левого верхнего пикселя для каждого отдельного участка размером $s \times s$ и отбрасывание остальных, а \mathbf{n} обычно считается *аддитивным белым гауссовым шумом* (additive white Gaussian noise, AWGN), определяемым стандартным отклонением (или уровнем шума) σ . Приведенная выше модель ухудшения является общей моделью для *сверхразрешения одиночного изображения* (single image superresolution, SISR). Однако если масштабный коэффициент s равен 1, она становится *моделью деградации изображения* (deblurring degradation model) с устранением размытия; дальнейшая фиксация \mathbf{k} как дельта-ядра превращает ее в *модель деградации с шумоподавлением* (denoising degradation model).

Поскольку IR – это некорректно поставленная обратная задача, для ограничения пространства решений необходимо принять априорную оценку, которую также называют *регуляризацией* (Roth, Black, 2009; Zoran, Weiss, 2011). С байесовской точки зрения решение $\hat{\mathbf{x}}$ может быть получено путем нахождения *апостериорного максимума* (maximum a posteriori, MAP):

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}), \quad (14.1)$$

где $\log p(\mathbf{y}|\mathbf{x})$ представляет собой логарифмическую вероятность наблюдения \mathbf{y} , $\log p(\mathbf{x})$ – априорное значение чистого изображения \mathbf{x} и не зависит от ухудшенного изображения \mathbf{y} . Более формально (14.1) можно переписать в виде:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k}) \downarrow_s\|^2 + \lambda \mathcal{R}(\mathbf{x}), \quad (14.2)$$

где решение минимизирует энергетическую функцию, состоящую из члена данных $\frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k}) \downarrow_s\|^2$ и члена регуляризации $\lambda \mathcal{R}(\mathbf{x})$ с параметром регуляризации λ . В частности, член данных гарантирует, что решение соответствует процессу деградации, в то время как член регуляризации смягчает некорректность задачи, навязывая желаемое свойство решению.

Как правило, методы решения уравнения (14.2) можно разделить на две основные категории, а именно методы, основанные на моделях, и методы, основанные на обучении. Первые направлены на непосредственное решение уравнения (14.2) с применением алгоритмов оптимизации, в то время как последние в основном изучают предопределенную параметризованную функцию путем оптимизации функции потерь на обучающем наборе, содержащем N пар ухудшенных/исходных изображений $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^N$ (Tappen, 2007; Barbu, 2009; Sun, Tappen, 2013; Schmidt, Roth, 2014; Chen, Pock, 2017). В частности, методы, основанные на обучении, обычно моделируются как следующая задача двухуровневой оптимизации:

$$\begin{cases} \min_{\theta} \sum_{i=1}^N \mathcal{L}(\hat{\mathbf{x}}_i, \mathbf{x}_i) \\ \text{так, что } \hat{\mathbf{x}}_i = f(\mathbf{y}_i, \theta) \end{cases}, \quad (14.3a)$$

$$(14.3b)$$

где θ обозначает обучаемые параметры, $\mathcal{L}(\hat{x}_i, x_i)$ измеряет потерю полученного чистого изображения \hat{x}_i по отношению к истинному изображению x_i .

Основное различие между методами, основанными на модели, и методами, основанными на обучении, заключается в том, что первые проявляют гибкость при решении различных задач IR, просто определяя операции ухудшения, и могут напрямую оптимизировать изображение y с ухудшенным качеством, тогда как вторые требуют трудоемкого обучения модели перед применением и обычно ограничиваются специализированными задачами. Тем не менее методы, основанные на обучении, могут не только обеспечивать высокое быстродействие, но и, как правило, обеспечивают лучшее качество благодаря сквозному обучению. В отличие от них, для обеспечения хорошего качества результата основанные на модели методы обычно нуждаются в больших затратах времени на подбор сложных априорных значений (Gu et al., 2014). Например, методы на основе моделей, такие как NCSR (Dong et al., 2013), гибко решают задачи обработки шумоподавления, суперразрешения и устранения размытия, тогда как методы на основе глубокого обучения MLP (Burger et al., 2012), SRCNN (Dong et al., 2016), DCNN (Xu et al., 2014) должны быть настроены только на решение конкретной задачи. И даже в случае конкретной задачи, такой как шумоподавление, методы на основе моделей (например, BM3D (Dabov et al., 2007) и WNNM (Gu et al., 2014)) могут гибко работать с разными уровнями шума, тогда как основанный на обучении метод (Jain, Seung, 2009) отдельно обучает разные модели для каждого уровня. Следовательно, эти две категории методов имеют свои достоинства и недостатки, и было бы полезно исследовать их интеграцию, чтобы объединить достоинства.

Интеграция методов, основанных на моделях и на обучении, привела к созданию метода IR на основе глубокого обучения с поддержкой PnP. Он заменяет подзадачу оптимизации шумоподавляющей модели шумоподавителем на основе предварительно обученной CNN. Основная идея глубокого PnP IR заключается в том, что с помощью алгоритмов переменного разделения, таких как *метод переменного направления множителей* (alternating direction method of multipliers, ADMM) (Boyd et al., 2011) и *полуквадратичное разделение* (half-quadratic splitting, HQS) (Geman, Yang, 1995), можно работать с членом данных и априорным членом регуляризации по отдельности (Parikh et al., 2014), когда, в частности, априорный член соответствует только подзадаче шумоподавления (Danielyan et al., 2010; Heide et al., 2014; Venkatakrishnan et al., 2013), которую можно решить с помощью глубокого шумоподавителя CNN. Следовательно, метод глубокого PnP можно рассматривать как частный случай методов, основанных на моделях.

Помимо PnP, интеграцию подходов моделирования и глубокого обучения можно также выполнить с помощью методов *глубокой развертки* (deep unfolding). Их основное отличие состоит в том, что последние оптимизируют параметры сквозным путем, минимизируя функцию потерь на большом обучающем наборе, и, таким образом, обычно дают лучшие результаты даже при меньшем количестве итераций. Математически метод глубокой развертки можно записать следующим образом:

$$\begin{cases} \min_{\theta} \sum_{i=1}^N \mathcal{L}(\hat{\mathbf{x}}_i, \mathbf{x}_i) \\ \text{так, что } \hat{\mathbf{x}}_i = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|\mathbf{y}_i - (\mathbf{x} \otimes \mathbf{k}) \downarrow_s\|^2 + \lambda \mathcal{R}(\mathbf{x}) \end{cases} \quad (14.4a)$$

$$(14.4b)$$

Другими словами, метод глубокой развертки направлен на замену предопределенной функции из уравнения (14.36) с разверткой уравнения вывода (14.46), поэтому его можно рассматривать как частный случай методов, основанных на обучении. Основная идея глубокой развертки применительно к задаче IR состоит в том, чтобы сначала развернуть основанную на модели функцию энергии с помощью алгоритма полуквадратичного разделения и получить вывод, который итеративно чередуется между решением двух подзадач, одна из которых связана с членом данных, а другая – с априорным членом, а затем рассматривать вывод как глубокую нейросеть, заменяя решения двух подзадач нейронными модулями. Поскольку две упомянутые подзадачи соответствуют получению знаний о согласованности деградации и априорных знаний шумоподавителя, метод глубокой развертки основан на явной деградации и априорных ограничениях, что является отличительным преимуществом по сравнению с простыми методами SISR, основанными на обучении. На рис. 14.1 показана иллюстрация связей между методами, основанными на глубоком обучении, методами, основанными на модели, методами глубокого PnP и глубокой развертки.

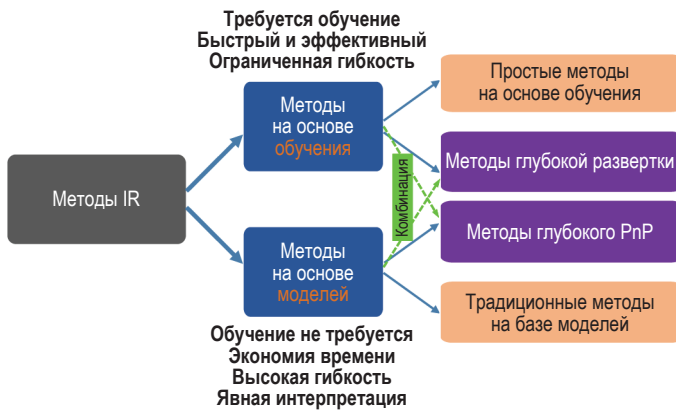


Рис. 14.1 ❖ Иллюстрация связей между методами глубокого обучения, моделирования, глубокого PnP и глубокой развертки

Остальная часть этой главы организована следующим образом. Поскольку методы глубокого PnP и глубокой развертки тесно связаны с алгоритмами переменного разделения, в разделе 14.2 мы сначала представим один из самых популярных, то есть алгоритм *полуквадратичного разделения* (half quadratic splitting, HQS). Затем в разделе 14.3 представим методы глубокого PnP. В частности, мы предлагаем очень гибкий и эффективный априорный шу-

моподавитель CNN, а затем подключаем его в качестве модуля к алгоритму HQS для решения различных проблем восстановления изображений. В разделе 14.4 мы рассмотрим методы восстановления изображения с глубокой разверткой, которые могут решать задачи устранения размытия и суперразрешения с различными ядрами размытия и коэффициентами масштабирования с помощью одной модели. В разделе 14.5 приведены количественные и качественные результаты, а также представлен тщательный анализ настройки гиперпараметров и промежуточных результатов, чтобы лучше понять механизм работы методов глубокого PnP и глубокой развертки.

14.2. АЛГОРИТМ ПОЛУКВАДРАТИЧНОГО РАЗДЕЛЕНИЯ (HQS)

Хотя существуют различные алгоритмы разделения переменных для решения уравнения (14.2), алгоритм полуквадратичного разделения (HQS) обязан своей популярностью простоте и быстрой сходимости. Поэтому в данной главе мы используем HQS. Как правило, HQS решает уравнение (14.2) введением вспомогательной переменной \mathbf{z} , что дает нам следующее приближенно эквивалентное уравнение:

$$E_\mu(\mathbf{x}, \mathbf{z}) = \frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k}) \downarrow_s\|^2 + \lambda \Phi(\mathbf{x}) + \frac{\mu}{2} \|\mathbf{z} - \mathbf{x}\|^2, \quad (14.5)$$

где μ – параметр штрафа. Такое уравнение можно решить путем итеративного решения промежуточных уравнений для \mathbf{x} и \mathbf{z}

$$\begin{cases} \mathbf{z}_k = \operatorname{argmin}_{\mathbf{z}} \|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k}) \downarrow_s\|^2 + \mu \sigma^2 \|\mathbf{z} - \mathbf{x}_{k-1}\|^2 \\ \mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}} \frac{\mu}{2} \|\mathbf{z}_k - \mathbf{x}\|^2 + \lambda \Phi(\mathbf{x}) \end{cases} \quad (14.6)$$

$$\quad (14.7)$$

Согласно уравнению (14.6), значение μ должно быть достаточно большим, чтобы \mathbf{x} и \mathbf{z} были приблизительно равны фиксированной точке. Однако это также приведет к медленной сходимости. Следовательно, хорошее эмпирическое правило состоит в том, чтобы итеративно увеличивать μ . Для удобства на k -й итерации μ обозначается как μ_k .

Можно заметить, что член данных и член регуляризации разделены на уравнения (14.6) и (14.7) соответственно. Для решения уравнения (14.6) можно использовать быстрое преобразование Фурье (БПФ), если предположить, что свертка выполняется с круговыми граничными условиями. Примечательно, что преобразование выражено в закрытой форме (Zhao et al., 2016):

$$\mathbf{z}_k = \mathcal{F}^{-1} \left(\frac{1}{\alpha_k} \left(\mathbf{d} - \overline{\mathcal{F}(\mathbf{k})} \odot_s \frac{(\mathcal{F}(\mathbf{k})\mathbf{d}) \downarrow_s}{(\mathcal{F}(\mathbf{k})\mathcal{F}(\mathbf{k})) \downarrow_s + \alpha_k} \right) \right). \quad (14.8)$$

В свою очередь, \mathbf{d} определяется как

$$\mathbf{d} = \overline{\mathcal{F}(\mathbf{k})} \mathcal{F}(\mathbf{y} \uparrow_s) + \alpha_k \mathcal{F}(\mathbf{x}_{k-1}),$$

где $\alpha_k \triangleq \mu_k \sigma^2$ и где $\mathcal{F}(\cdot)$ и $\mathcal{F}^{-1}(\cdot)$ обозначают БПФ и обратное БПФ, $\overline{\mathcal{F}(\cdot)}$ обозначает комплексное сопряжение $\mathcal{F}(\cdot)$; \odot_s – оператор обработки отдельных блоков с поэлементным умножением, т. е. применение поэлементного умножения к $\mathbf{s} \times \mathbf{s}$ различных блоков $\overline{\mathcal{F}(\mathbf{k})}$; \downarrow_s обозначает понижение разрешения (downsampling) отдельных блоков, т. е. усреднение по $\mathbf{s} \times \mathbf{s}$ блокам; \uparrow_s обозначает стандартную s -кратную повышающую дискретизацию, т. е. повышение пространственного разрешения путем заполнения новых записей нулями. Для частного случая устранения размытия, когда $\mathbf{s} = 1$, уравнение (14.8) можно кратко записать как

$$\mathbf{z}_k = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\mathbf{k})} \mathcal{F}(\mathbf{y}) + \alpha_k \mathcal{F}(\mathbf{x}_{k-1})}{\overline{\mathcal{F}(\mathbf{k})} \mathcal{F}(\mathbf{k}) + \alpha_k} \right). \quad (14.9)$$

Другими словами, уравнение (14.8) обобщает уравнение (14.9). Для решения уравнения (14.7) полезно знать, что с байесовской точки зрения оно фактически соответствует задаче шумоподавления с уровнем шума $\beta_k \triangleq \sqrt{\lambda/\mu_k}$ (Chan et al., 2017).

14.3. ГЛУБОКОЕ ВОССТАНОВЛЕНИЕ ИЗОБРАЖЕНИЯ ПО МЕТОДУ PnP

Восстановление изображения по методу PnP обычно состоит из двух этапов. Первый шаг заключается в разделении члена данных и члена регуляризации целевой функции с помощью какого-либо алгоритма разделения переменных, что приводит к итеративной схеме, состоящей из поочередного решения подзадачи данных и подзадачи априорного распределения. Второй шаг – решить подзадачу априорного распределения с помощью любых готовых шумоподавителей, таких как K-SVD (Elad, Aharon, 2006), нелокального среднего (Buades et al., 2005), BM3D (Dabov et al., 2007). В результате, в отличие от традиционных методов, основанных на моделях, которые должны содержать явные и созданные вручную априорные значения, IR PnP может неявно определять их с помощью шумоподавителя. Такое преимущество дает возможность использовать очень глубокий шумоподаватель CNN для повышения качества результата.

IR PnP можно проследить до работ (Danielyan et al., 2010; Zoran, Weiss, 2011; Venkatakrisnan et al., 2013). Даниелян и др. (2012) использовали равновесие Нэша для реализации метода итеративного устранения размытия BM3D (IDDBM3D). В (Egiazarian, Katkovnik, 2015) аналогичный метод, предварительно оснащенный шумоподавитель CBM3D, был предложен для решения задачи повышения разрешения одиночного изображения (SISR). Благодаря

итеративному обновлению шага обратной проекции и шага шумоподавления CBM3D метод демонстрирует обнадеживающие перспективы улучшения показателя PSNR¹ по сравнению с SRCNN (Dong et al., 2016). В более ранней работе Даниеляна и др. (2010), в задаче устранения размытия изображения для объединения шумоподавателя BM3D, был применен расширенный метод Лагранжа. В работе (Venkatakrishnan et al., 2013) была предложена итерационная схема, аналогичная (Danielyan et al., 2012) – первой работе, в которой шумоподаватель рассматривается как PnP. До этого аналогичная идея PnP упоминается в (Zoran, Weiss, 2011), где алгоритм HQS используется для шумоподавления, устранения размытия и раскрашивания изображения. Хайде и др. (Heide et al., 2014) использовали альтернативу ADMM и HQS, т. е. первично-двойственный алгоритм (Chambolle, Pock, 2011), чтобы отделить член данных от члена регуляризации. Теодоро и др. (Teodoro et al., 2016) добавили класс-ориентированный шумоподаватель модели гауссовой смеси (GMM) (Zoran, Weiss, 2011) в ADMM для устранения размытости изображения и сжатия изображения. Метцлер и др. (Metzler et al., 2016) разработали метод приближенной передачи сообщений с шумоподавлением (AMP) для интеграции шумоподавателей, таких как BLS-GSM (Portilla et al., 2003) и BM3D, в схему восстановления сжатых данных. Чан и др. (Chan et al., 2017) предложили алгоритм ADMM PnP с шумоподавателем BM3D для сверхразрешения одиночного изображения и восстановления квантованного изображения Пуассона. Камиллов и др. (Kamilov et al., 2017) предложили пороговый алгоритм быстрой итерационной усадки (FISTA) с шумоподавателями BM3D и WNNM (Gu et al., 2014) для нелинейного обратного рассеяния. Сан и др. (Sun et al., 2019) предложили FISTA, предварительно подключив TV и шумоподаватель BM3D для птихографической микроскопии Фурье. Яир и Михаэли (Yair, Michaeli, 2018) предложили использовать шумоподаватель WNNM в качестве готового решения для раскрашивания и удаления размытия. Гаваскар и Чаудхури (Gavaskar, Chaudhury, 2020) исследовали конвергенцию IR PnP на основе ISTA с шумоподавлением по алгоритму нелокального среднего.

С развитием методов глубокого обучения, таких как современные архитектуры сети и алгоритм оптимизации на основе градиента, шумоподаватель на основе CNN показал многообещающие характеристики с точки зрения качества и экономичности. После его успеха было предложено множество работ по тематике IR PnP на основе шумоподавателя CNN. Романо и др. (Romano et al., 2017) предложили явную регуляризацию с помощью шумоподавателя TNRD для устранения размытия изображения. В нашей предыдущей работе (Zhang et al., 2017) различные шумоподаватели CNN обучены подключаться к алгоритму HQS для устранения размытия и SISR. Тирер и Гириес (Tirer and Giryes, 2018) предложили применять итеративное шумоподавление и обратное проецирование с помощью шумоподавателей IRCNN для раскрашивания и устранения размытия изображения. Гу и др. (Gu et al., 2018) предложили использовать шумоподаватели WNNM и IRCNN для удаления размытия по принципу PnP и SISR. Тирер и Гириес (Tirer, Giryes, 2019) предложили использовать шумоподаватели IRCNN для SISR с поддержкой PnP. Ли и Ву (Li, Wu,

¹ Peak signal-to-noise ratio – пиковое отношение сигнала к шуму. – *Прим. перев.*

2019) подключили шумоподаватели IRCNN к алгоритму разделенной итерации Брегмана, чтобы решить задачу прорисовки глубины изображения. Рю и др. (Ryu et al., 2019) представили теоретический анализ сходимости IR PnP на основе алгоритмов прямого-обратного разделения и ADMM, а также предложили спектральную нормализацию для обучения шумоподавателя DnCNN. Сан и др. (Sun et al., 2019) разработали алгоритм регуляризации блочных координат путем шумоподавления (RED) с использованием шумоподавателя DnCNN (Zhang et al., 2017) в качестве явного регуляризатора.

Хотя PnP IR может использовать мощные ресурсы шумоподавателя CNN, существующие методы обычно основаны на шумоподавателях DnCNN или IRCNN, которые не в полной мере используют CNN. Как правило, шумоподаватель для PnP IR не должен быть слепым и должен справляться с широким диапазоном уровней шума. Однако шумоподавитель DnCNN необходимо обучить отдельную модель для каждого уровня шума. Чтобы уменьшить количество шумоподавателей, в некоторых работах используется один шумоподаватель, настроенный на небольшой уровень шума. Однако, по данным (Romano et al., 2017), такая стратегия, как правило, требует большого количества итераций для достижения удовлетворительного качества, что увеличивает вычислительную нагрузку. Хотя шумоподаватели IRCNN могут обрабатывать широкий диапазон уровней шума, они состоят из 25 отдельных 7-слойных шумоподавателей, и каждый из них обучается на интервальном уровне шума 2. Такой шумоподаватель страдает от следующих двух недостатков. Во-первых, у него нет гибкости в обработке разных уровней шума. Во-вторых, он недостаточно эффективен из-за неглубоких слоев. Учитывая вышеизложенное, необходимо разработать гибкий и мощный шумоподаватель для повышения показателей IR PnP. Предложенный нами шумоподаватель на основе FFDNet (Zhang et al., 2018a) может обрабатывать широкий диапазон уровней шума с помощью одной модели, используя в качестве входных данных карту уровня шума. Более того, его эффективность повышается за счет использования как ResNet (He et al., 2016), так и U-Net (Ronneberger et al., 2015). Глубокий шумоподаватель дополнительно включен в архитектуру PnP IR на основе HQS, чтобы продемонстрировать преимущества использования мощного глубокого шумоподавателя. В то же время предлагается новый периодический *геометрический самоансамбль* (geometric self-ensemble) для потенциального улучшения производительности без дополнительных вычислительных затрат, а также проводится тщательный анализ настройки параметров и промежуточных результатов, чтобы лучше понять механизм работы предлагаемого глубокого PnP IR.

14.3.1. Предварительное изучение глубокого шумоподавателя CNN

Хотя недавно были предложены различные методы шумоподавления на основе CNN, большинство из них не предназначены для IR с поддержкой PnP. В исследованиях (Lehtinen et al., 2018; Krull et al., 2019; Batson, Royer, 2019)

предлагается новая стратегия обучения без реальных данных. В (Guo et al., 2019; Brooks et al., 2019; Abdelhamed et al., 2019; Zamir et al., 2020) предложен метод синтеза реального шума для имитации реальных цифровых фотографий. Однако с байесовской точки зрения шумоподавитель для PnP IR должен быть гауссовым шумоподавителем. Следовательно, для обучения с учителем можно добавить синтетический гауссов шум к чистому изображению. В работах (Lefkimmiatis, 2017; Zhang et al., 2019; Liu et al., 2018; Plötz, Roth, 2018) для лучшего восстановления изображения в структуру сети был включен нелокальный модуль. Однако эти методы изучают отдельную модель для каждого уровня шума. Возможно, наиболее подходящим шумоподавителем для PnP IR является FFDNet (Zhang et al., 2018), который может обрабатывать широкий диапазон уровней шума, используя карту уровня шума в качестве входных данных. Тем не менее FFDNet демонстрирует показатели, сравнимые только с DnCNN и IRCNN, поэтому ей не хватает внутреннего потенциала для повышения качества PnP IR. По этой причине мы предлагаем улучшить архитектуру FFDNet, воспользовавшись преимуществами широко используемых U-Net (Ronneberger et al., 2015) и ResNet (He et al., 2016).

14.3.1.1. Шумоподавляющая сетевая архитектура

Хорошо известно, что U-Net (Ronneberger et al., 2015) эффективно и качественно преобразовывает изображение в изображение, в то время как ResNet (He et al., 2016) лучше подходит для увеличения качества моделирования за счет стека нескольких обходных блоков (residual block). Продолжая идею FFDNet (Zhang et al., 2018), которая использует карту уровня шума в качестве входных данных, предлагаемый шумоподавитель DRUNet дополнительно интегрирует обходные блоки в U-Net для эффективного моделирования априорного компонента шумоподавителя. Стоит подчеркнуть, что эта работа не направлена на разработку новой сетевой архитектуры шумоподавления. Схожую идею объединения U-Net и ResNet можно найти и в других работах, таких как (Zhang et al., 2018; Venkatesh et al., 2018).

Как и FFDNet, сеть DRUNet может обрабатывать различные уровни шума с помощью одной модели. Основой DRUNet является U-Net, состоящая из четырех градаций. Каждая градация имеет связь с пропуском идентичности между операциями шаговой свертки 2×2 (stride convolution, SConv) понижения разрешения и транспонированной свертки 2×2 (transposed convolution, TConv) повышения разрешения. Количество каналов в каждом слое от первой до четвертой градации составляет 64, 128, 256 и 512 соответственно. В каждой градации уменьшения и увеличения разрешения задействовано четыре последовательных обходных блока. По аналогии с устройством сетевой архитектуры для суперразрешения в (Lim et al., 2017) функции активации не следуют за первым и последним сверточными (Conv) слоями, а также слоями SConv и TConv. Кроме того, каждый обходной блок содержит только одну функцию активации ReLU.

Стоит отметить, что рассматриваемая DRUNet не имеет смещения, что означает, что смещение не используется во всех слоях Conv, SConv и TConv.

Причина двоякая. Во-первых, несмещенная сеть с активацией ReLU и пропуском идентичности естественным образом обеспечивает свойство инвариантности масштабирования многих задач восстановления изображений, т. е. условие $f(ax) = af(x)$ выполняется для любого скаляра $a \geq 0$ (см. Mohan et al., 2019 для более подробной информации). Во-вторых, мы эмпирически обнаружили, что для сети со смещением величина смещения будет намного больше, чем у фильтров, что, в свою очередь, может повредить обобщаемости.

14.3.2. Методика обучения

Хорошо известно, что CNN выигрывает от наличия крупномасштабных обучающих данных. Чтобы обогатить априорные данные шумоподавителя для PnP IR, вместо обучения на небольшом наборе данных, который включает 400 изображений набора данных сегментации Беркли (BSD) размером 180×180 (Chen, Pock, 2017), мы создаем большой набор данных, состоящий из 400 изображений BSD, 4744 изображений из исследовательской базы данных Waterloo (Ma et al., 2017), 900 изображений из набора данных DIV2K (Agustsson, Timofte, 2017) и 2750 изображений из набора данных Flickr2K (Lim et al., 2017). Поскольку такой набор данных охватывает большее пространство изображений, обученная модель может немного улучшить показатель PSNR для набора данных BSD68 (Roth, Black, 2009), имея при этом очевидный выигрыш в PSNR при обработке наборов данных из другого домена.

В соответствии с общепринятой методикой для гауссова шумоподавления зашумленный аналог y чистого изображения x получается путем добавления AWGN с уровнем шума σ . Соответственно, карта уровня шума представляет собой однородную карту, заполненную значениями σ , и имеет тот же пространственный размер, что и зашумленное изображение. Чтобы справиться с широким диапазоном уровней шума, во время обучения уровень шума σ выбирается случайным образом из диапазона $[0, 50]$. Параметры сети оптимизируются путем минимизации потерь L_1 , а не L_2 между изображением после шумоподавления и его истинным образцом с помощью алгоритма Адама (Kingma, Ba, 2015). Хотя нет прямых доказательств того, какая потеря приведет к повышению качества, широко признано, что потеря L_1 более робастна, чем потеря L_2 при обработке выбросов (Bishop, 2006). Что касается шумоподавления, во время выборки AWGN могут возникать выбросы. В этом смысле при обучении шумоподавляющей сети потери L_1 имеют тенденцию быть более стабильными, чем потери L_2 . Шаг обучения начинается с $1 \cdot 10^{-4}$, затем уменьшается вдвое каждые 100 000 итераций, и, наконец, обучение заканчивается, когда шаг становится меньше $5 \cdot 10^{-7}$. На каждой итерации во время обучения из обучающих данных случайным образом отбирались 16 патчей размером 128×128 . Мы отдельно обучаем модель шумоподавителя для изображений в градациях серого и цветных изображений. Обучение модели с помощью PyTorch и графического процессора Nvidia Titan Xp занимает около четырех дней.

14.3.3. Результаты удаления шума

14.3.3.1. Удаление шума с изображений в градациях серого

Для очистки от шума изображений в градациях серого мы сравнили предложенный нами шумоподаватель DRUNet с несколькими современными методами удаления шума, включая два репрезентативных метода на основе моделей – BM3D (Dabov et al., 2007) и WNNM (Gu et al., 2014), один метод на основе CNN, который отдельно обучает одну модель для каждого уровня шума (например, DnCNN (Zhang et al., 2017)), и два метода на основе CNN, которые были обучены работать с широким диапазоном уровней шума (например, IRCNN (Zhang et al., 2017) и FFDNet (Zhang et al., 2018)). Значения PSNR различных методов на широко используемых наборах данных Set12 (Zhang et al., 2017) и BSD68 (Martin et al., 2001; Roth, Black, 2009) для уровней шума 15, 25 и 50 представлены в табл. 14.1. Можно видеть, что DRUNet достигает наилучших показателей PSNR для всех уровней шума в двух наборах данных. В частности, DRUNet дает среднее увеличение PSNR около 0,9 дБ по сравнению с BM3D и превосходит DnCNN, IRCNN и FFDNet по среднему значению PSNR 0,5 дБ в наборе данных Set12 и 0,25 дБ в наборе данных BSD68. На рис. 14.2 показаны результаты очистки изображения в градациях серого различными методами на примере снимка бабочки Монарх из набора данных Set12 с уровнем шума 50. Видно, что DRUNet может восстанавливать гораздо более четкие края, чем DnCNN и FFDNet.

Таблица 14.1. Средние значения PSNR (дБ) для различных методов с уровнями шума 15, 25 и 50 дБ на популярных наборах данных Set12 и BSD68 (Martin et al., 2001; Roth, Black, 2009; Zhang et al., 2017). Лучшие результаты выделены жирным шрифтом

Набор	Уровень шума	BM3D	WNNM	DnCNN	IRCNN	FFDNet	DRUNet
Set12	15	32,37	32,70	32,86	32,77	32,75	33,25
	25	29,97	30,28	30,44	30,38	30,43	30,94
	50	26,72	27,05	27,18	27,14	27,32	27,90
BSD68	15	31,08	31,37	31,73	31,63	31,63	31,91
	25	28,57	28,83	29,23	29,15	29,19	29,48
	50	25,60	25,87	26,23	26,19	26,29	26,59

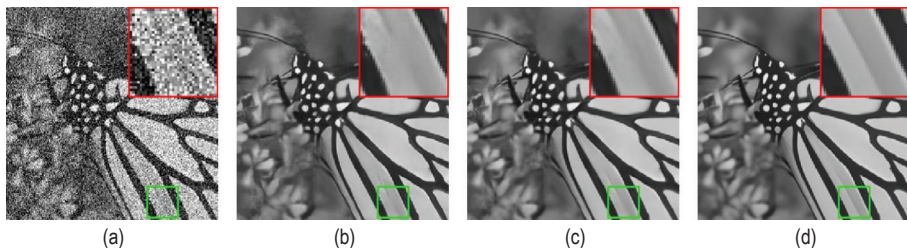


Рис. 14.2 ❖ Результаты очистки от шума изображения в градациях серого различными методами на примере снимка бабочки Монарх из набора данных Set12 с уровнем шума 50. а) Шум (14,78 дБ); б) DnCNN (26,83 дБ); в) FFDNet (26,92 дБ); д) DRUNet (27,31 дБ)

14.3.3.2. Удаление шума с цветного изображения

Поскольку существующие методы в основном сосредоточены на очистке от шума изображений в градациях серого, при подавлении шумов на цветном изображении мы сравниваем DRUNet только с CBM3D, DnCNN, IRCNN и FFDNet. В табл. 14.2 представлены результаты очистки цветного изображения с использованием различных методов для уровней шума 15, 25 и 50 на CBSD68 (Martin et al., 2001; Roth, Black, 2009; Zhang et al., 2017), Kodak24 (Franzen, 1999) и McMaster (Zhang et al., 2011). Можно видеть, что DRUNet значительно превосходит другие конкурирующие методы. Стоит отметить, что, имея хорошие показатели в наборе данных CBSD68, DnCNN хуже работает с набором данных McMaster. Такое несоответствие подчеркивает важность сокращения разрыва между обучением и применением в области подавления шумов на изображении. Примеры применения различных методов к изображению «163085» из набора данных CBSD68 с уровнем шума 50 показаны на рис. 14.3, по которому видно, что DRUNet может восстановить больше мелких деталей и текстур, чем конкурирующие методы.



Рис. 14.3 ❖ Результаты шумоподавления цветного изображения различными методами на изображении «163085» из набора данных CBSD68 с уровнем шума 50. а) Исходное зашумленное изображение (14,99 дБ); б) DnCNN (28,68 дБ); в) FFDNet (28,75 дБ); г) DRUNet (29,28 дБ)

Таблица 14.2. Средние значения PSNR (дБ), полученные различными методами для уровней шума 15, 25 и 50 дБ на наборах CBSD68 (Martin et al., 2001; Roth, Black, 2009; Zhang et al., 2017a), Kodak24 и McMaster. Лучшие результаты выделены жирным шрифтом

Набор	Уровень шума	CBM3D	DnCNN	IRCNN	FFDNet	DRUNet
CBSD68	15	33,52	33,90	33,86	33,87	34,30
	25	30,71	31,24	31,16	31,21	31,69
	50	27,38	27,95	27,86	27,96	28,51
Kodak24	15	34,28	34,60	34,69	34,63	35,31
	25	32,15	32,14	32,18	32,13	32,89
	50	28,46	28,95	28,93	28,98	29,86
McMaster	15	34,06	33,45	34,58	34,66	35,40
	25	31,66	31,52	32,18	32,35	33,14
	50	28,51	28,62	28,91	29,18	30,08

14.3.4. Алгоритм HQS для PnP IR

Как упоминалось ранее, мы выбрали HQS в качестве алгоритма разделения переменных из-за его простоты и быстрой сходимости. Между тем не вызывает сомнений, что настройка параметров всегда является нетривиальной задачей (Romano et al., 2017). Другими словами, для получения хорошего качества обработки изображений необходима тщательная настройка параметров. Чтобы лучше понять принцип PnP IR на основе HQS, мы обсудим общую методологию настройки параметров после краткого обзора алгоритма HQS. Затем рассмотрим стратегию периодического геометрического самосогласованного ансамбля, позволяющую улучшить качество.

14.3.4.1. Алгоритм полуквадратичного разделения (HQS)

HQS использует следующую итеративную схему для решения уравнения (14.2):

$$\begin{cases} \mathbf{x}_k = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k}) \downarrow_s\|^2 + \mu \sigma^2 \|\mathbf{x} - \mathbf{z}_{k-1}\|^2 \end{cases} \quad (14.10a)$$

$$\begin{cases} \mathbf{z}_k = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{2(\sqrt{\lambda/\mu_k})^2} \|\mathbf{z} - \mathbf{x}_k\|^2 + \mathcal{R}(\mathbf{z}) \end{cases} \quad (14.10b)$$

В частности, подзадача (14.10b) с байесовской точки зрения соответствует гауссову шумоподавлению на \mathbf{x}_k с уровнем шума $\beta_k \triangleq \sqrt{\lambda/\mu_k}$. Следовательно, любой гауссов шумоподавитель может быть подключен к чередующимся итерациям для решения (14.2). Чтобы решить это уравнение, мы перепишем (14.10b) следующим образом:

$$\mathbf{z}_k = \text{Шумоподавитель}(\mathbf{x}_k, \beta_k). \quad (14.11)$$

Из уравнения (14.11) можно сделать два вывода. Во-первых, приор $\mathcal{R}(\cdot)$ может быть неявно задан шумоподавителем. По этой причине как приор, так и шумоподавитель для IR PnP обычно называют априорным шумоподавителем. Во-вторых, интересно обучить единственный шумоподавитель CNN, чтобы заменить уравнение (14.11) и использовать преимущества CNN, такие как высокая гибкость проектирования сети, высокая эффективность графических процессоров и мощные возможности моделирования с глубокими сетями.

14.3.4.2. Общая методика настройки параметров

Из чередующихся итераций между уравнением (14.10a) и уравнением (14.10b) легко видеть, что задействованы три регулируемых параметра, включая параметр штрафа μ , параметр регуляризации λ и общее количество итераций K . Как правило, чтобы гарантировать, что \mathbf{x}_k и \mathbf{z}_k сходятся к фиксированной точке, требуется большое значение μ , что, однако, требует большого K для сходимости. Общий подход состоит в стратегии постепенного увеличения μ ,

что приводит к последовательности $\mu_1 < \dots < \mu_k < \dots < \mu_K$. Тем не менее в таком случае необходимо ввести новый параметр для управления размером шага, что усложняет настройку параметров. Согласно уравнению (14.11), мы можем заметить, что μ определяет уровень шума $\beta_k = \sqrt{\lambda/\mu_k}$ на k -й итерации априорного шумоподавителя. С другой стороны, предполагается, что диапазон уровня шума $[0, 50]$ достаточен для β_k . Вдохновленные таким знанием о домене, мы можем настроить β_k и λ для неявного определения μ_k . Исходя из того, что значение μ_k должно монотонно возрастать, мы равномерно выбираем β_k от большого уровня шума β_1 до малого β_K в логарифмическом пространстве. Это означает, что μ_k легко определяется через $\mu_k = \lambda/\beta_k^2$. По аналогии с (Zhang et al., 2017) мы фиксируем β_1 на уровне 49, а β_K определяется уровнем шума изображения β . Поскольку K задается пользователем, а σ_K имеет ясный физический смысл, их легко настроить на практике. Фактически нам осталось настроить параметр λ . Поскольку λ происходит от члена регуляризации и, следовательно, должен быть фиксированным, мы можем выбрать оптимальное значение λ с помощью поиска по сетке в проверочном наборе данных. Эмпирически установлено, что λ обеспечивает необходимое качество в диапазоне $[0,19, 0,55]$. В этой главе мы используем значение 0,23, если не указано иное. Следует отметить, что поскольку λ может быть поглощен β и играет роль контроля компромисса между членом данных и членом регуляризации, можно неявно настроить λ , умножив β на скаляр.

14.3.4.3. Периодический геометрический самосогласованный ансамбль

Геометрический ансамбль, основанный на отражении и вращении, является широко используемой стратегией для повышения качества IR (Timofte et al., 2016). Сначала он преобразует входные данные путем отражения и поворота для создания 8 изображений, затем получает соответствующие восстановленные изображения и, наконец, выдает усредненный результат после обратного преобразования. Хотя использование подобного геометрического ансамбля дает определенный выигрыш в качестве, он достигается за счет увеличения времени вывода.

В отличие от описанного выше метода, мы периодически применяем геометрический ансамбль для каждой последовательных 8 итераций. На каждой итерации этот процесс включает одно преобразование перед шумоподавлением и соответствующее обратное преобразование после шумоподавления. Обратите внимание, что мы отказались от шага усреднения, потому что вход априорной модели шумоподавителя варьируется в зависимости от итераций. Мы называем этот метод периодическим геометрическим самосогласованным ансамблем. Его явное преимущество заключается в том, что общее время вывода не увеличивается. Эмпирически мы обнаружили, что предложенный метод обычно может улучшить PSNR на 0,02 ~ 0,2 дБ.

Основываясь на вышеизложенном, мы оформили стратегию глубокого PnP IR, а именно DPIR, в виде обобщенного алгоритма 1.

Алгоритм 1. Восстановление изображения по принципу PnP с предварительным глубоким шумоподавлением (DPIR)

Вход: априорная модель глубокого шумоподавителя, зашумленное изображение \mathbf{y} , модель деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}$, уровень шума изображения σ , β_k априорной модели шумоподавителя на k -й итерации из K итераций, компромиссный параметр λ .

Выход: Восстановленное изображение \mathbf{z}_K .

1. Инициализируем \mathbf{z}_0 из \mathbf{y} , предварительно вычисляем $\alpha_k = \lambda \sigma^2 / \beta_k^2$
 2. **for** $k = 1, 2, \dots, K$ **do**
 3. $\mathbf{x}_k = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k}) \downarrow_s\|^2 + \alpha_k \|\mathbf{x} - \mathbf{z}_{k-1}\|^2$; // Решение подзадачи данных
 4. $\mathbf{z}_k = \text{Шумоподаватель}(\mathbf{x}_k, \beta_k)$; // Шумоподавление с помощью глубокого шумоподавителя DRUNet и периодического геометрического ансамбля
 5. **end**
-

14.4. ВОССТАНОВЛЕНИЕ ИЗОБРАЖЕНИЯ МЕТОДОМ ГЛУБОКОЙ РАЗВЕРТКИ

Первые методы глубокой развертки восходят к исследованиям (Barbu, 2009; Samuel, Tappen, 2009; Sun, Tappen, 2011), где для очистки изображения от шума был предложен компактный вывод MAP, основанный на алгоритме градиентного спуска. С тех пор были предложены различные методы глубокой развертки, основанные на определенных алгоритмах оптимизации (например, полуквадратичное разделение (Afonso et al., 2010), метод переменного направления множителей (Boyd et al., 2011) и прямо-двойственный метод (Chambolle and Pock, 2011)) и предназначенные для решения различных задач восстановления изображений, таких как подавление шумов (Chen, Pock, 2017; Lefkimmiatis, 2017), устранение размытости (Schmidt, Roth, 2014; Kruse et al., 2017) и компрессионное зондирование (Zhang, Ghanem, 2018). По сравнению с простыми методами, основанными на обучении, методы глубокой развертки поддаются интерпретации и могут включать в модель ограничения деградации. Однако большинство из них страдают одним или несколькими из следующих недостатков. (1) Решение подзадачи регуляризации без использования глубокой CNN является недостаточно мощным для хорошего качества. (2) Подзадача данных не имеет решения в закрытой форме, что может препятствовать сходимости. (3) Весь вывод обучается поэтапно и с тонкой настройкой, а не полным сквозным способом. Поэтому особый интерес вызывает метод, не страдающий вышеупомянутыми недостатками.

14.4.1. Сеть глубокой развертки

После определения оптимизации развертки, т. е. алгоритма HQS, следующим шагом является проектирование сети *восстановления изображений с глубокой разверткой* (deep unfolding image restoration, DUIR). Поскольку оптимизация развертки в основном состоит из итеративного решения подзадачи данных в виде уравнения (14.6) и подзадачи регуляризации в виде уравнения (14.7), DUIR должен переключаться между модулем данных \mathcal{D} и модулем приора \mathcal{P} . Кроме того, поскольку решения подзадач также принимают на вход соответствующие гиперпараметры α_k и β_k , в DUIR дополнительно вводится модуль гиперпараметров \mathcal{H} . На рис. 14.4 изображена общая архитектура DUIR с K итераций, где значение K эмпирически выбрано равным 8 для достижения компромисса между скоростью и качеством. Далее приводится более подробная информация о \mathcal{D} , \mathcal{P} и \mathcal{H} .

14.4.1.1. Модуль данных \mathcal{D}

Модуль данных играет роль уравнения (14.8) которое является решением подзадачи данных в закрытой форме. По сути, он предназначен для поиска более четкого изображения с высоким разрешением, которое минимизирует взвешенную комбинацию члена данных $\|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k})\|_s^2$ и члена квадратичной регуляризации $\|\mathbf{z} - \mathbf{x}_{k-1}\|^2$ с компромиссным гиперпараметром α_k . Поскольку член данных соответствует модели деградации, модуль данных не только имеет то преимущество, что принимает масштабный коэффициент \mathbf{s} и ядро размытия \mathbf{k} в качестве входных данных, но также накладывает ограничение деградации на решение. На самом деле сложно вручную разработать такой простой, но полезный модуль с несколькими входами. Для краткости перепишем уравнение (14.8) следующим образом:

$$\mathbf{z}_k = \mathcal{D}(\mathbf{x}_{k-1}, \mathbf{s}, \mathbf{k}, \mathbf{y}, \alpha_k). \quad (14.12)$$

Обратите внимание, что \mathbf{x}_0 инициализируется путем простой интерполяции \mathbf{y} по ближайшему соседу с масштабным коэффициентом \mathbf{s} . Следует отметить, что уравнение (14.12) не содержит обучаемых параметров, что, в свою очередь, приводит к лучшей обобщаемости из-за полного разделения между членом данных и членом регуляризации. Для реализации модуля мы используем PyTorch, где основные операторы БПФ и обратного БПФ могут быть реализованы с помощью `torch.fft.fftn` и `torch.fft.ifftn` соответственно.

14.4.1.2. Модуль приора \mathcal{P}

Модуль приора предназначен для получения более чистой оценки \mathbf{x}_k путем пропуска \mathbf{z}_k через шумоподаватель, когда уровень шума равен β_k . По аналогии с (Zhang et al., 2018) мы предлагаем использовать глубокий шумоподаватель CNN, который принимает уровень шума в качестве входных данных:

$$\mathbf{x}_k = \mathcal{P}(\mathbf{z}_k, \beta_k). \quad (14.13)$$

Предлагаемый шумоподаватель, а именно ResUNet, объединяет обходные блоки (He et al., 2016) в U-Net (Ronneberger et al., 2015). U-Net широко используется для сравнения изображений, в то время как ResNet обязана своей популярностью быстрому обучению и большой емкости со множеством обходных блоков.

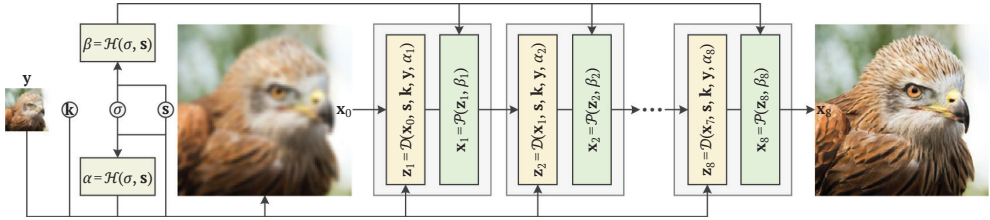


Рис. 14.4 ❖ Общая архитектура DUIR с $K = 8$ итерациями. DUIR может гибко обрабатывать выход модели деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}$ с помощью одной модели, поскольку в качестве входных данных она принимает искаженное изображение \mathbf{y} , масштабный коэффициент \mathbf{s} , ядро размытия \mathbf{k} и уровень шума σ . В частности, DUIR состоит из трех основных модулей, включая модуль данных \mathcal{D} , который делает изображение высокого разрешения более четким, модуль приора \mathcal{P} , который делает изображение более чистым, и модуль гиперпараметров \mathcal{H} , управляющий выходными данными \mathcal{D} и \mathcal{P}

ResUNet принимает объединенную карту \mathbf{z}_k и уровня шума в качестве входных данных и выводит очищенное от шума изображение \mathbf{x}_k . Таким образом, ResUNet может обрабатывать различные уровни шума с помощью одной модели, что значительно сокращает общее количество параметров. Следуя общей структуре U-Net, ResUNet содержит четыре масштабных пути, каждый из которых имеет сквозную связь с пропуском идентичности между операциями уменьшения и увеличения масштаба. В частности, количество каналов в слоях с первого по четвертый масштабный путь установлено равным 64, 128, 256 и 512 соответственно. Для операций понижения и повышения разрешения используются пошаговая свертка 2×2 (SConv) и транспонированная свертка 2×2 (TConv) соответственно. Заметим также, что за слоями SConv и TConv, а также за первым и последним сверточными слоями не следуют функции активации. Ради наследования достоинств ResNet при уменьшении и увеличении разрешения применяется группа из двух обходных блоков. Как было предложено в (Lim et al., 2017), каждый обходной блок состоит из двух сверточных слоев 3×3 с активацией ReLU в середине и сквозным соединением с пропуском идентичности, суммируемым с его выходом.

14.4.1.3. Модуль гиперпараметров \mathcal{H}

Модуль гиперпараметров действует как «ползунковый регулятор» для управления выводом модулей данных и приора. Например, решение \mathbf{z}_k будет постепенно приближаться к \mathbf{x}_{k-1} по мере увеличения α_k . Согласно определению α_k и β_k , α_k зависит от σ и μ_k , а β_k зависит от λ и μ_k . Хотя можно найти фиксиро-

рованные λ и μ_k , мы утверждаем, что возможно улучшение качества результата, если λ и μ_k меняются в зависимости от двух ключевых элементов, т. е. масштабного коэффициента \mathbf{s} и уровня шума σ , которые влияют на степень некорректности задачи. Пусть $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_K]$ и $\beta = [\beta_1, \beta_2, \dots, \beta_K]$: мы используем единственный модуль для предсказания α и β :

$$[\alpha, \beta] = \mathcal{H}(\sigma, \mathbf{s}). \quad (14.14)$$

Модуль гиперпараметров состоит из трех полностью связанных слоев с ReLU в качестве первых двух функций активации и Softplus в качестве последней. Количество скрытых узлов в каждом слое равно 64. Учитывая тот факт, что α_k и β_k должны быть положительными, а уравнению (14.8) следует избегать деления на чрезвычайно малые α_k , к выходу слоя Softplus дополнительно прибавляют 1×10^{-6} .

14.4.2. Сквозное обучение

Сквозное обучение направлено на изучение параметров DUIR путем минимизации функции потерь на большом наборе обучающих данных. Поэтому в данном разделе в основном описываются обучающие данные, функция потерь и методика обучения. Вслед за (Wang et al., 2018) мы используем DIV2K (Agustsson, Timofte, 2017) и Flickr2K (Timofte et al., 2017). Ухудшенные изображения синтезируются с помощью модели деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_{\mathbf{s}} + \mathbf{n}$. Масштабные коэффициенты выбираются из множества $\{1, 2, 3, 4\}$. В качестве ядер размытия мы используем анизотропные ядра Гаусса, как в (Riegler et al., 2015; Shocher et al., 2018; Zhang et al., 2018), и ядра движения, как в (Boracchi, Foi, 2012). Мы также используем фиксированный размер ядра 25×25 . Диапазон уровня шума принят равным $[0, 25]$. Что касается функции потерь, для оценки величины PSNR мы используем потери L_1 . Для оптимизации параметров DUIR применяем решатель Adam (Kingma, Ba, 2015) с размером мини-пакета 128. Коэффициент обучения начинается с 1×10^{-4} , уменьшается в 0,5 раза каждые 4×10^4 итераций и, наконец, заканчивается на 3×10^{-6} . Стоит отметить, что из-за невозможности параллельных вычислений для разных коэффициентов масштабирования каждый мини-пакет включает только один случайный коэффициент масштабирования. Размер патча чистого образа равен 96×96 . Мы обучаем модели с помощью PyTorch на четырех графических процессорах Nvidia Tesla V100 в облаке Amazon AWS. Получение модели DUIR занимает около двух дней.

14.5. ЭКСПЕРИМЕНТЫ

Чтобы проверить гибкость и эффективность DPIR и DUIR, мы рассматриваем две классические задачи IR – устранение размытия изображения и сверхразрешение одиночного изображения (SISR). Для каждой задачи мы продемонстрируем количественные и качественные результаты DPIR и DUIR на трех

классических тестовых изображениях, которые показаны на рис. 14.5. Мы сравним разработанную ручную и изученную настройку гиперпараметров между DPIR и DUIR. Кроме того, проведем визуальное сравнение \mathbf{x}_k и \mathbf{z}_k на промежуточных итерациях для DPIR и DUIR.

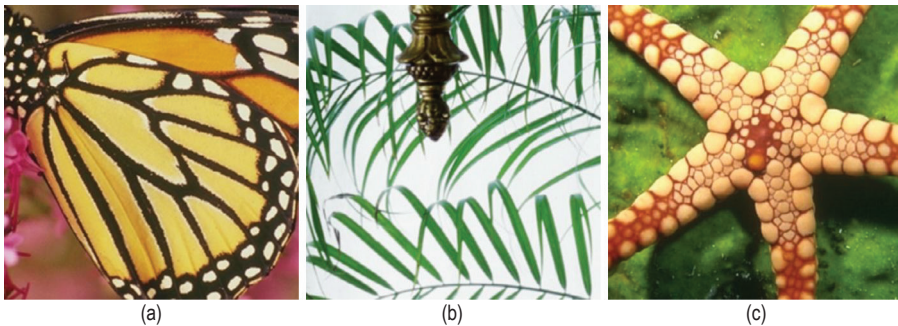


Рис. 14.5 ❖ Три классических тестовых изображения:
(a) бабочка; (b) листья; (c) морская звезда

14.5.1. Устранение размытия изображения

Мы используем для тестирования два из восьми ядер размытия, предложенных в (Levin et al., 2009), которые имеют размер 17×17 и 27×27 соответственно. Как показано в табл. 14.3, мы также рассматриваем гауссов шум с различными уровнями шума: 2,55 (1 %) и 7,65 (3 %). Следуя общей стратегии равномерного устранения размытия, мы синтезируем размытые изображения, сначала применяя ядро размытия, а затем добавляя AWGN с уровнем шума σ . Что касается гиперпараметров DPIR, то K и σ_K устанавливаются равными 8 и σ соответственно, а \mathbf{z}_0 инициализируется как \mathbf{y} .

Таблица 14.3. Значения PSNR (дБ) архитектур DPIR и DUIR в задаче устранения размытия трех тестовых изображений. Лучшие результаты выделены жирным шрифтом

Метод	σ	Бабочка	Листья	Морская звезда
Второе ядро размером 17×17 из (Levin et al., 2009)				
DPIR	2,55	34,26	35,19	34,21
DUIR		33,73	34,35	34,02
DPIR	7,65	29,52	30,11	29,83
DUIR		29,55	30,02	29,84
Четвертое ядро размером 27×27 из (Levin et al., 2009)				
DPIR	2,55	34,18	35,12	33,91
DUIR		33,58	34,26	33,73
DPIR	7,65	29,45	30,27	29,46
DUIR		29,38	29,94	29,43

14.5.1.1. Количественные и качественные результаты

В табл. 14.3 представлены значения PSNR (дБ) архитектур DPIR и DUIR для трех тестовых изображений. Видно, что DPIR и DUIR достигают схожих результатов. Обратите внимание, что DPIR имеет более крупный и медленный шумоподаватель, чем DUIR, поэтому DPIR менее эффективен, чем DUIR. Визуальные результаты DPIR и DUIR после обработки тестовых изображений показаны на рис. 14.6. Можно видеть, что и DPIR, и DUIR могут эффективно восстанавливать резкость и естественный вид изображения.

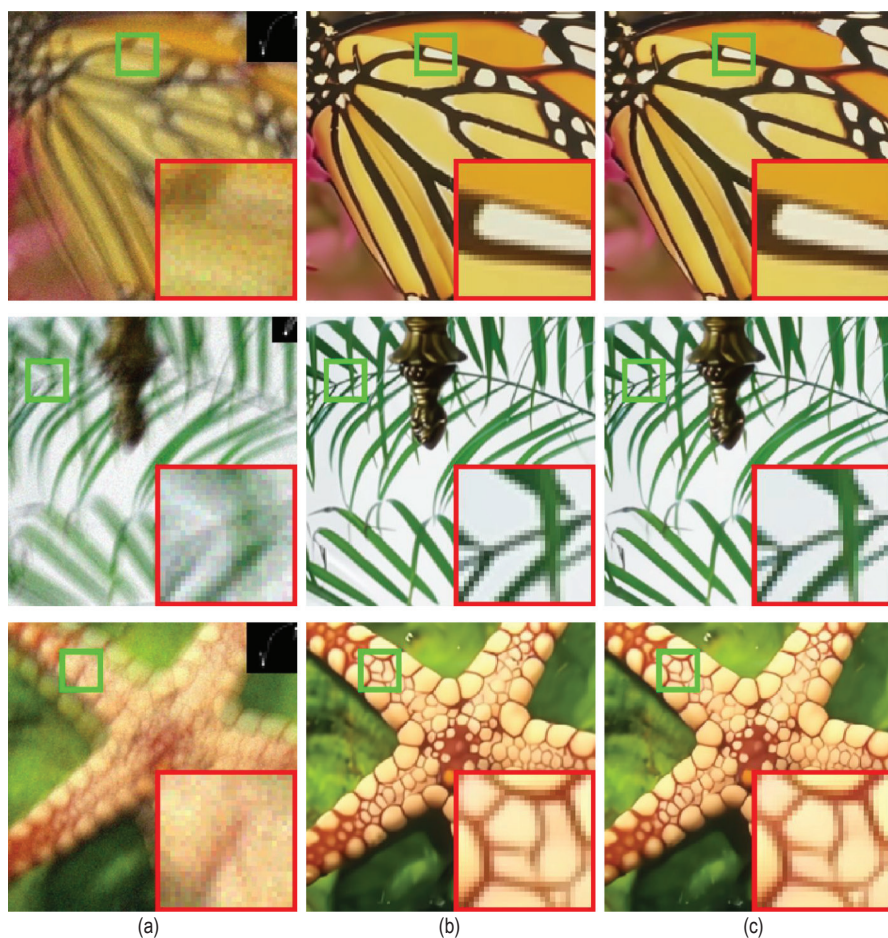


Рис. 14.6 ❖ Визуальные результаты DPIR и DUIR в задаче устранения размытия изображения. Ядро размытия показано в правом верхнем углу размытого изображения. Уровень шума 7,65 (3 %). (a) Размытое изображение; (b) DPIR; (c) DUIR

14.5.1.2. Сравнение найденных вручную и обученных гиперпараметров

Рисунки 14.7 и 14.8 иллюстрируют гиперпараметры, то есть α и β , для DPIR и DUIR соответственно. Можно видеть, что изученные гиперпараметры DUIR

в целом соответствуют разработанным вручную гиперпараметрам DPIR. По рис. 14.7 и 14.8а видно, что α положительно коррелирует с σ . По рис. 14.7 и 14.8b видно, что β имеет тенденцию к уменьшению с количеством итераций и увеличивается с уровнем шума. Это означает, что уровень шума промежуточной оценки постепенно снижается на протяжении итераций, а серьезное ухудшение требует большого β_k , чтобы справиться с некорректностью.

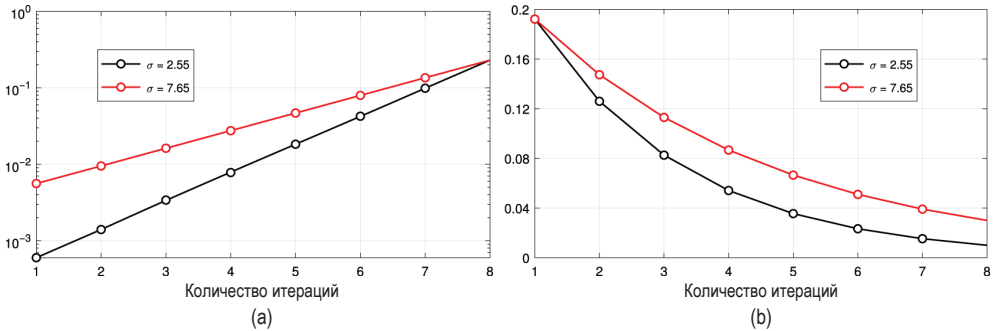


Рис. 14.7 ❖ Гиперпараметры а) α и б) β DPIR в задаче устранения размытия по отношению к различным уровням шума

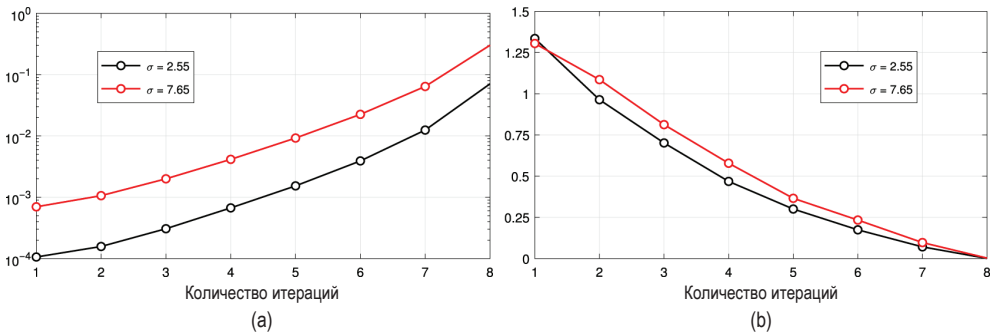


Рис. 14.8 ❖ Гиперпараметры а) α и б) β DUIR в задаче устранения размытия по отношению к различным уровням шума

14.5.1.3. Промежуточные результаты

На рис. 14.9 и 14.10 изображены визуальные результаты и PSNR для \mathbf{x}_k и \mathbf{z}_k при различных итерациях DPIR и DUIR на изображении морской звезды с (14.12). По рис. 14.9 видно, что хотя решение в замкнутой форме \mathbf{x}_1 может справиться с искажением размытия, оно также усугубляет шум. Глубокий шумоподаватель удаляет шум, что дает нам \mathbf{z}_k без шума. По мере увеличения числа итераций \mathbf{x}_7 содержит меньше структурированного шума, чем \mathbf{x}_1 , а \mathbf{z}_7 восстанавливает больше деталей и более четкие края, чем \mathbf{z}_1 . По рис. 14.10 видно, что \mathcal{D} и \mathcal{P} могут облегчать друг другу итеративное и попеременное удаление размытия. Интересно, что \mathbf{z}_1 DPIR значительно отличается от \mathbf{z}_1 DUIR, а это означает, что, в отличие от DPIR, априорный шумоподаватель в DUIR не является априорным гауссовым шумоподавателем.

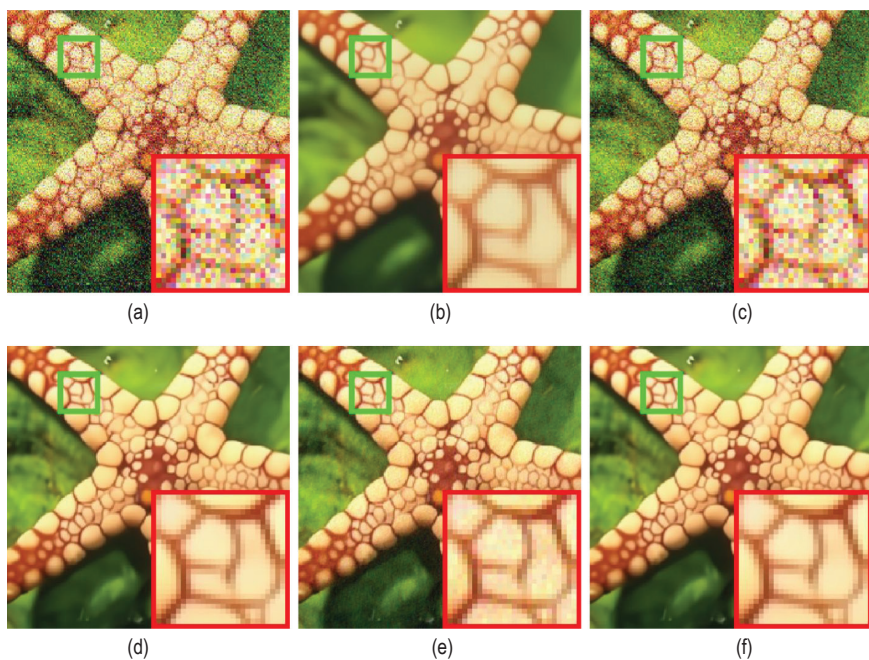


Рис. 14.9 ❖ Оценки в различных итерациях DPIR для задачи удаления размытия изображения морской звезды на рис. 14.6. (a) \mathbf{x}_1 ; (b) \mathbf{z}_1 ; (c) \mathbf{x}_2 ; (d) \mathbf{z}_6 ; (e) \mathbf{x}_7 ; (f) \mathbf{z}_7

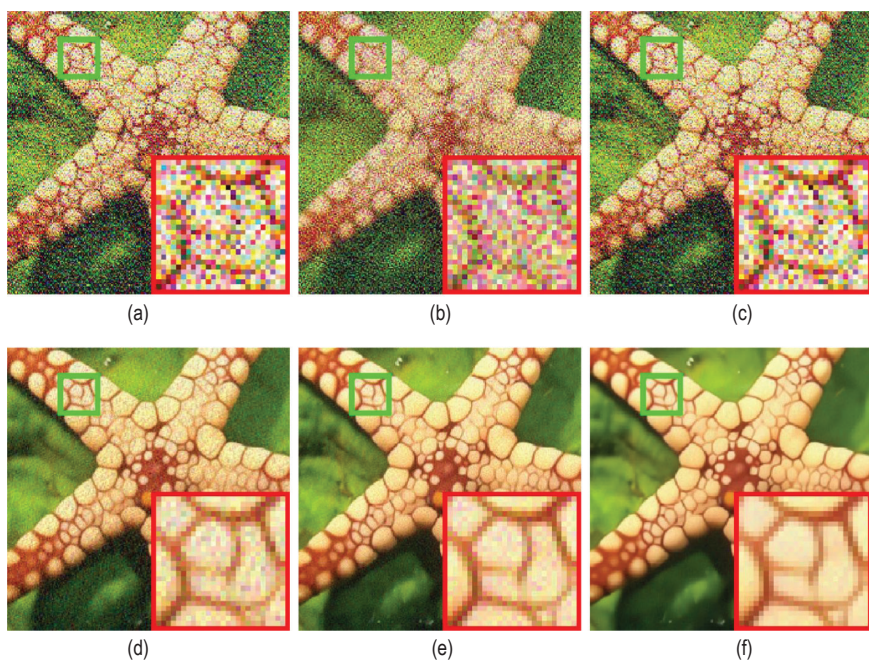


Рис. 14.10 ❖ Оценки в различных итерациях DUIR для задачи удаления размытия изображения морской звезды на рис. 14.6. (a) \mathbf{x}_1 ; (b) \mathbf{z}_1 ; (c) \mathbf{x}_2 ; (d) \mathbf{z}_6 ; (e) \mathbf{x}_7 ; (f) \mathbf{z}_7

Мы можем сделать следующие выводы. Во-первых, хотя уравнение (14.10a) может справиться с искажением размытия, оно также усугубляет влияние шума по сравнению с входом \mathbf{z}_{k-1} . Во-вторых, априорный глубокий шумоподаватель успешно удаляет шум, что дает нам очищенный от шума \mathbf{z}_k . В-третьих, по сравнению с \mathbf{x}_1 и \mathbf{x}_2 , \mathbf{x}_8 содержит больше мелких деталей, а это означает, что уравнение (14.10a) может итеративно восстанавливать детали.

14.5.2. Сверхразрешение одиночного изображения (SISR)

Хотя мы используем модель деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}$ для SISR, стоит отметить, что существующие методы SISR в основном предназначены для бикубической модели деградации с формулой $\mathbf{y} = \mathbf{x} \downarrow_s^{\text{bicubic}}$, где $\downarrow_s^{\text{bicubic}}$ обозначает бикубическую модель даунсэмплинга с понижающим коэффициентом s . Однако было обнаружено, что качество этих методов серьезно ухудшается, если реальная модель деградации отклоняется от предполагаемой (Efrat et al., 2013; Zhang et al., 2015). Поскольку бикубическая деградация хорошо изучена, интересно исследовать ее связь с классической моделью деградации. На самом деле бикубическую деградацию можно аппроксимировать, установив соответствующее ядро размытия в модели деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s$. Чтобы достичь этого, мы используем метод работы с большими данными. Он заключается в решении следующей задачи оценки ядра путем минимизации ошибки реконструкции для большой пары высокое разрешение / бикубическое низкое разрешение $\{(\mathbf{x}, \mathbf{y})\}$:

$$\mathbf{k}_{\text{bicubic}}^{*s} = \operatorname{argmin}_{\mathbf{k}} \|(\mathbf{x} \otimes \mathbf{k}) \downarrow_s - \mathbf{y}\|. \quad (14.15)$$

На рис. 14.11 показаны аппроксимированные бикубические ядра для масштабных коэффициентов 2, 3 и 4. Следует отметить, что поскольку операция понижающей дискретизации выбирает верхний левый пиксель для каждого отдельного фрагмента $\mathbf{s} \times \mathbf{s}$, бикубические ядра для масштабных коэффициентов 2, 3 и 4 имеют смещение центра на 0,5, 1 и 1,5 пикселя в направлении вверх влево соответственно.

Для синтеза соответствующих тестовых изображений низкого разрешения с помощью модели деградации $\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}$ необходимо задать ядра размытия и уровни шума. Для более тщательной оценки было бы полезно использовать большое количество ядер размытия и уровней шума; однако это также приведет к обременительному процессу оценки. По этой причине мы рассматриваем только 8 репрезентативных и разнообразных ядер размытия, в том числе 4 изотропных ядра Гаусса с разной шириной (т. е. 0,7, 1,2, 1,6 и 2,0) и 4 анизотропных ядра Гаусса из (Zhang et al., 2018б). Мы не рассматриваем ядра размытия в движении, поскольку было указано, что для задачи SISR достаточно ядер Гаусса. Таким образом, для дальнейшего анализа робастности ядра мы будем сообщать результаты PSNR отдельно для каждого ядра размытия, а не для каждого типа ядра. Хотя существует мнение, что правильное ядро размытия должно варьироваться в зависимости от коэффициента

масштабирования (Zhang et al., 2015), мы утверждаем, что 8 ядер размытия достаточно разнообразны, чтобы покрыть большое пространство ядер. Для уровней шума мы выбираем значения 2,55 (1 %) и 7,65 (3 %). Общие параметры K и σ_K устанавливаются равными 24 и $\max(\sigma, s)$ соответственно. Для инициализации z_0 используется бикубическая интерполяция изображения низкого разрешения. В частности, поскольку классическая модель деградации выбирает верхний левый пиксель для каждого отдельного фрагмента $s \times s$, необходимо надлежащим образом решить проблему сдвига. Чтобы решить эту проблему, мы настраиваем z_0 с помощью интерполяции сетки.

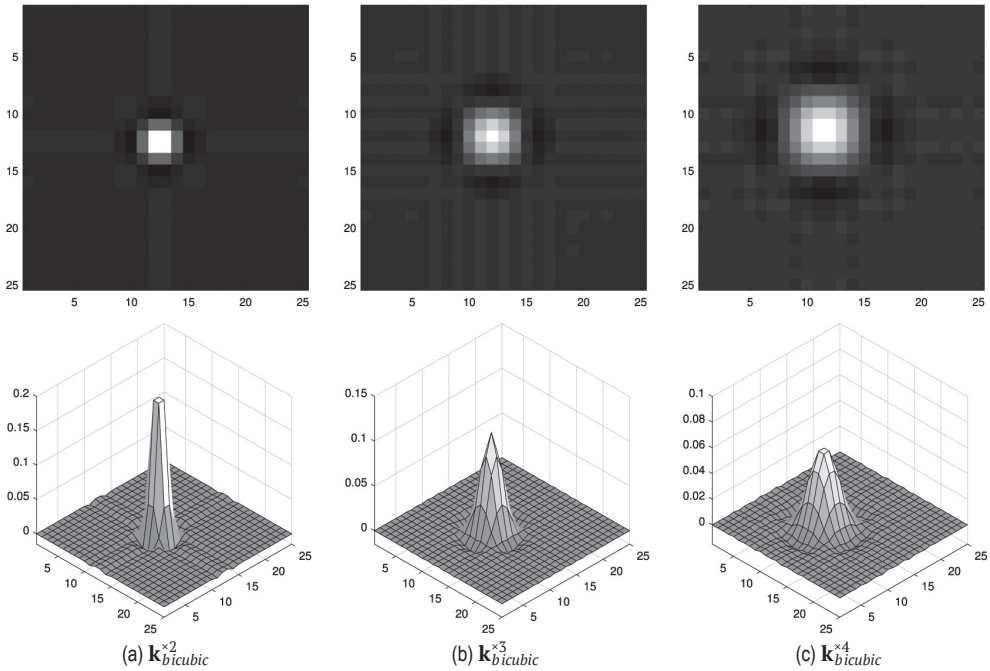


Рис. 14.11 Аппроксимированные бикубические ядра для коэффициентов масштабирования 2, 3 и 4 в рамках модели деградации $y = (x \otimes k) \downarrow_s$. Обратите внимание, что эти ядра содержат отрицательные значения

14.5.2.1. Количественное и качественное сравнение

В табл. 14.4 представлены средние значения PSNR (дБ) для DPIR и DUIR для трех тестовых изображений. Из табл. 14.4 видно, что значения PSNR различаются для разных ядер размытия, и к наивысшему PSNR для каждого коэффициента масштабирования приводят разные ядра. Точнее, больший коэффициент масштабирования обычно требует более плавного ядра размытия.

Кроме того, DUIR значительно превосходит DPIR. Такое явление указывает на то, что сквозное обучение более полезно для суперразрешения, чем для устранения размытия. На рис. 14.12 показаны визуальные результаты работы DPIR и DUIR на трех тестовых изображениях. Можно заметить, что как DPIR, так и DUIR могут значительно улучшить качество изображения. Заметим, что DUIR дает более четкие края, чем DPIR. Возможно, это связано со сквозным обучением.

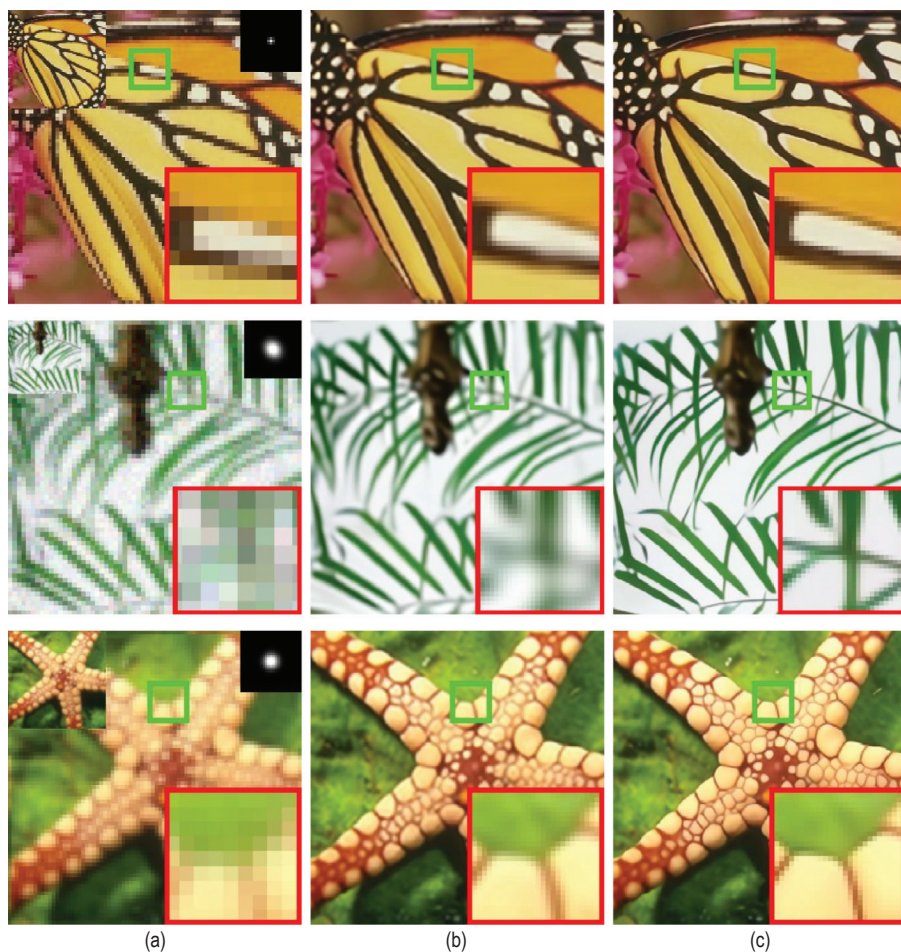










Рис. 14.12 ❖ Визуальные результаты работы DPIR и DUIR на трех тестовых изображениях. Ядро размытия показано в правом верхнем углу изображения низкого разрешения. (a) Изображение низкого разрешения; (b) DPIR; (c) DUIR

Таблица 14.4. Средние значения PSNR (дБ) для DPIR и DUIR для различных комбинаций коэффициентов масштабирования, ядер размытия и уровней шума. Лучшие результаты выделены жирным шрифтом

Метод	Коэффициент масштабирования	Уровень шума	Ядро размытия							
										
DPIR	×2	0	31,79	32,10	31,28	29,63	28,72	28,62	29,45	27,77
	×3	0	26,04	26,82	26,97	26,89	26,73	25,89	26,58	26,49
	×3	2,55	25,91	26,49	26,23	25,43	24,81	24,44	25,29	24,13
	×3	7,65	25,53	25,67	24,91	23,68	23,14	22,81	23,56	22,33
	×4	0	22,62	23,50	23,74	23,85	23,76	23,18	23,58	23,88
DUIR	×2	0	33,58	34,47	33,49	31,79	31,29	31,29	31,45	30,20
	×3	0	28,05	29,53	29,88	29,87	29,40	29,37	29,43	29,25
	×3	2,55	27,77	28,74	28,47	27,70	27,15	27,20	27,42	26,54
	×3	7,65	26,87	27,13	26,43	25,47	25,15	25,03	25,24	24,47
	×4	0	24,71	26,38	27,02	27,30	27,16	26,76	26,69	27,28

14.5.2.2. Сравнение найденных вручную и изученных гиперпараметров

Рисунки 14.13 и 14.14 иллюстрируют гиперпараметры α и β сетей DPIR и DUIR для различных комбинаций масштабного коэффициента s и уровня шума σ соответственно. Можно видеть, что изученные гиперпараметры DUIR согласуются с найденными вручную гиперпараметрами DPIR. По рис. 14.13 и 14.14а видно, что α положительно коррелирует с σ и меняется с s . Это фактически согласуется с определением α_k . По рис. 14.13 и 14.14б видно, что β имеет тенденцию к уменьшению с ростом количества итераций и увеличивается вместе с масштабным коэффициентом и уровнем шума. Это означает, что уровень шума при получении высокого разрешения постепенно снижается по мере прохождения итераций, а комплексная деградация требует большого β_k для устранения некорректности задачи.

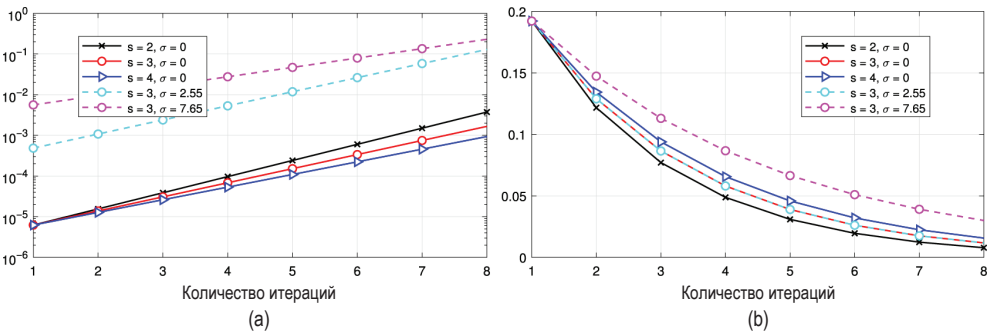


Рис. 14.13 ❖ Гиперпараметры (а) α и (б) β сети DPIR для задачи сверхразрешения при различных комбинациях уровней шума и масштабных коэффициентов

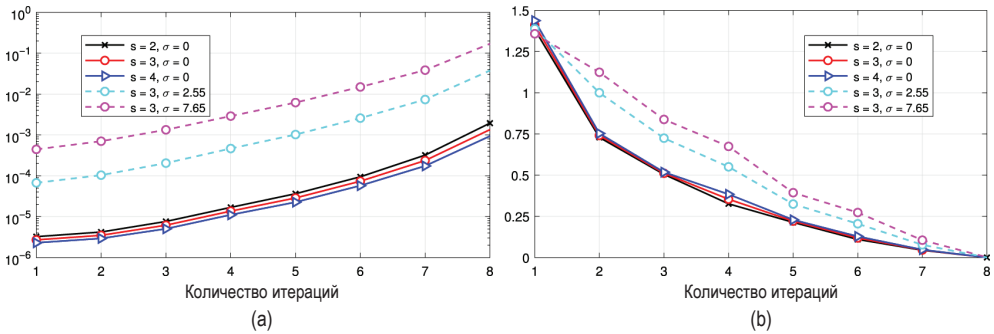


Рис. 14.14 ❖ Гиперпараметры (а) α и (б) β сети DUIR для задачи сверхразрешения при различных комбинациях уровней шума и масштабных коэффициентов

14.5.2.3. Промежуточные результаты

На рис. 14.15 и 14.16 представлены визуальные результаты \mathbf{x}_k и \mathbf{z}_k при различных итерациях DPIR и DUIR на примере изображения морской звезды с рис. 14.12. По рис. 14.15 видно, что, хотя изображение низкого разрешения не содержит шума, решение в замкнутой форме \mathbf{x}_1 вносит сильный структурированный шум. После прохождения \mathbf{x}_1 через гауссов шумоподавитель

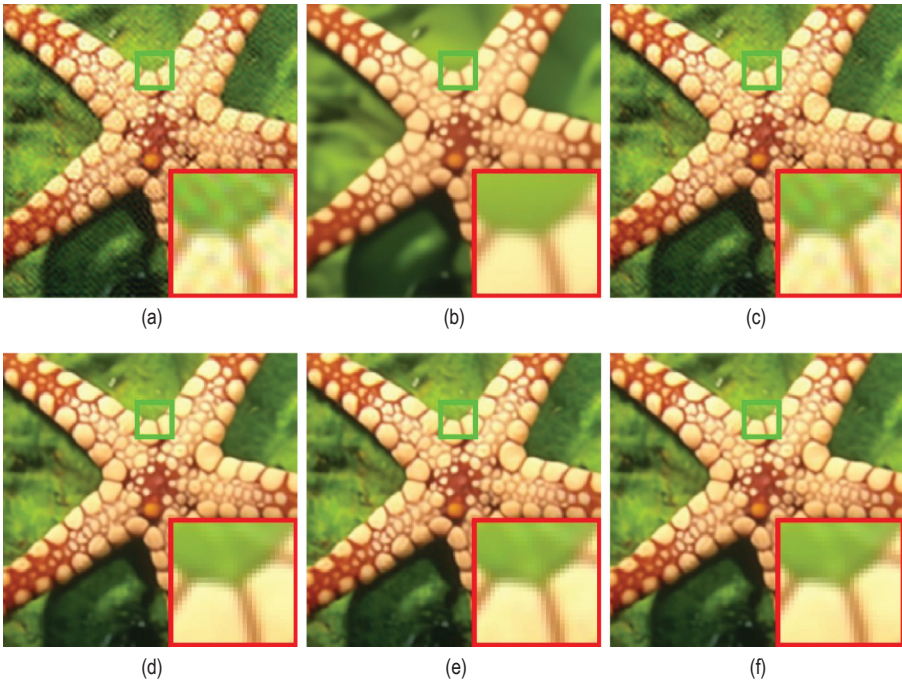


Рис. 14.15 ❖ Оценки задачи восстановления высокого разрешения в различных итерациях DPIR на примере изображения морской звезды с рис. 14.12. (а) \mathbf{x}_1 ; (б) \mathbf{z}_1 ; (с) \mathbf{x}_2 ; (д) \mathbf{z}_6 ; (е) \mathbf{x}_7 ; (ф) \mathbf{z}_7

такой структурированный шум удаляется, что видно из \mathbf{z}_1 . При этом крошечные текстуры и структуры сглаживаются, а края становятся размытыми. По мере увеличения числа итераций \mathbf{x}_7 содержит меньше структурированного шума, чем \mathbf{x}_1 , а \mathbf{z}_7 восстанавливает больше деталей и более четкие края, чем \mathbf{z}_1 . По рис. 14.16 видно, что \mathcal{D} и \mathcal{P} могут помогать друг другу в итеративном и попеременном удалении размытия и восстановлении деталей. Интересно, что, в отличие от гауссова шумоподавителя DPIR, \mathcal{P} также может действовать как усилитель детализации для высокочастотного восстановления, что, возможно, является следствием обучения для конкретной задачи. Кроме того, это не уменьшает деградацию, вызванную ядром размытия, что подтверждает развязку между \mathcal{D} и \mathcal{P} . В результате DUIR со сквозным обучением имеет определенное преимущество перед DPIR, зависящее от задачи.

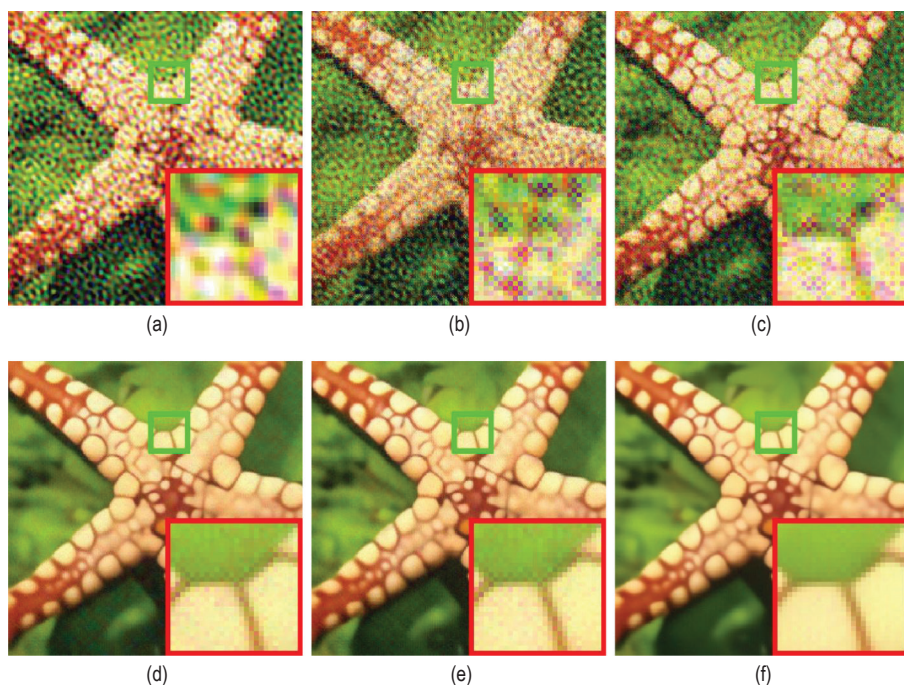


Рис. 14.16 ❖ Оценки задачи восстановления высокого разрешения в различных итерациях DUIR на примере изображения морской звезды с рис. 14.12. (a) \mathbf{x}_1 ; (b) \mathbf{z}_1 ; (c) \mathbf{x}_2 ; (d) \mathbf{z}_6 ; (e) \mathbf{x}_7 ; (f) \mathbf{z}_7

14.6. ЗАКЛЮЧЕНИЕ

В этой главе было рассказано о методах глубокого PnP и глубокой развертки, которые могут интегрировать методы, основанные на моделях и обучении, в задачу восстановления изображений. В частности, с помощью алгоритма полуквадратичного разделения, который может отделить член данных от

члена регуляризации, методы глубокого PnP могут заменить подзадачу регуляризации априорным шумоподавителем на основе глубокого обучения, в то время как методы глубокой развертки обучают итеративную сеть решать подзадачу регуляризации с помощью нейронных модулей. В результате как методы глубокого PnP, так и методы глубокой развертки могут унаследовать гибкость методов, основанных на моделях, сохраняя при этом преимущества методов, основанных на обучении. Для лучшего понимания механизма работы обоих разновидностей метода в главе представлены количественные и качественные результаты, а также сравнительный анализ настройки гиперпараметров и промежуточные результаты. Результаты показывают, что, хотя методы глубокой развертки изучают гиперпараметры, аналогичные разработанным вручную методам глубокого PnP, обученные априорные модули отличаются от априорных модулей гауссова шумоподавителя. Например, обученный априорный модуль методов глубокой развертки также может улучшить детализацию изображения, в то время как априорный гауссов шумоподаватель методов глубокого PnP не имеет такого достоинства.

Хотя предложенные методы глубокого PnP и глубокой развертки показали большие перспективы, на практике они также имеют несколько недостатков. Во-первых, это неслепые методы, которые требуют точной оценки параметров деградации, таких как ядро размытия. Если расчетное ядро размытия сильно отклоняется от истинного ядра, качество восстановления серьезно ухудшится. Во-вторых, они созданы на основе идеальной модели деградации, которая редко соответствует реальным изображениям. Если реальный шум не подчиняется аддитивному белому распределению Гаусса, это может привести к снижению качества.

БЛАГОДАРНОСТИ

Эта работа была частично поддержана фондом ETH Zürich Fund (OK) и проектом Huawei Technologies Oy (Финляндия).

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Abdelhamed A., Brubaker M. A., Brown M. S.*, 2019. Noise flow: noise modeling with conditional normalizing flows. In: IEEE International Conference on Computer Vision, pp. 3165–3173.
- Afonso M. V., Bioucas-Dias J. M., Figueiredo M. A.*, 2010. Fast image recovery using variable splitting and constrained optimization. IEEE Transactions on Image Processing 19 (9), 2345–2356.
- Agustsson E., Timofte R.*, 2017. NTIRE 2017 challenge on single image super-resolution: dataset and study. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, vol. 3, pp. 126–135.
- Andrews H. C., Hunt B. R.*, 1977. Digital Image Restoration. Prentice-Hall Signal Processing Series, vol. 1. Prentice-Hall, Englewood Cliffs.

- Barbu A.*, 2009. Training an active random field for real-time image denoising. *IEEE Transactions on Image Processing* 18 (11), 2451–2462.
- Batson J., Royer L.*, 2019. Noise2self: blind denoising by self-supervision. In: *International Conference on Machine Learning*, pp. 524–533.
- Bishop C. M.*, 2006. *Pattern Recognition and Machine Learning*. Springer.
- Boracchi G., Foi A.*, 2012. Modeling the performance of image restoration from motion blur. *IEEE TIP* 21 (8), 3502–3517.
- Boyd S., Parikh N., Chu E., Peleato B., Eckstein J.*, 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3 (1), 1–122.
- Brooks T., Mildenhall B., Xue T., Chen J., Sharlet D., Barron J. T.*, 2019. Unprocessing images for learned raw denoising. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11,036–11,045.
- Buades A., Coll B., Morel J. M.*, 2005. A non-local algorithm for image denoising. In: *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65.
- Burger H. C., Schuler C. J., Harmeling S.*, 2012. Image denoising: can plain neural networks compete with BM3D? In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399.
- Chambolle A., Pock T.*, 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* 40 (1), 120–145.
- Chan S. H., Wang X., Elgendy O. A.*, 2017. PnP ADMM for image restoration: fixed-point convergence and applications. *IEEE Transactions on Computational Imaging* 3 (1), 84–98.
- Chen Y., Pock T.*, 2017. Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (6), 1256–1272.
- Dabov K., Foi A., Katkovnik V., Egiazarian K.*, 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing* 16 (8), 2080–2095.
- Danielyan A., Katkovnik V., Egiazarian K.*, 2010. Image deblurring by augmented Lagrangian with BM3D frame prior. In: *Workshop on Information Theoretic Methods in Science and Engineering*, pp. 16–18.
- Danielyan A., Katkovnik V., Egiazarian K.*, 2012. BM3D frames and variational image deblurring. *IEEE Transactions on Image Processing* 21 (4), 1715–1728.
- Dong C., Loy C. C., He K., Tang X.*, 2016. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (2), 295–307.
- Dong W., Zhang L., Shi G., Li X.*, 2013. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing* 22 (4), 1620–1630.
- Efrat N., Glasner D., Apartsin A., Nadler B., Levin A.*, 2013. Accurate blur models vs. image priors in single image super-resolution. In: *IEEE International Conference on Computer Vision*, pp. 2832–2839.
- Egiazarian K., Katkovnik V.*, 2015. Single image super-resolution via BM3D sparse coding. In: *European Signal Processing Conference*, pp. 2849–2853.

- Elad M., Aharon M.*, 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* 15 (12), 3736–3745.
- Franzen R.*, 1999. Kodak lossless true color image suite. source. <http://r0k.us/graphics/kodak>. vol. 4.
- Gavaskar R. G., Chaudhury K. N.*, 2020. PnP ista converges with kernel denoisers. *IEEE Signal Processing Letters* 27, 610–614.
- Geman D., Yang C.*, 1995. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing* 4 (7), 932–946.
- Gu S., Timofte R., Van Gool L.*, 2018. Integrating local and non-local denoiser priors for image restoration. In: *International Conference on Pattern Recognition*.
- Gu S., Zhang L., Zuo W., Feng X.*, 2014. Weighted nuclear norm minimization with application to image denoising. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862–2869.
- Guo S., Yan Z., Zhang K., Zuo W., Zhang L.*, 2019. Toward convolutional blind denoising of real photographs. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1712–1722.
- He K., Zhang X., Ren S., Sun J.*, 2016. Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- Heide F., Steinberger M., Tsai Y. T., Rouf M., Pajak D., Reddy D., Gallo O., Liu J., Heidrich W., Egiazarian K., et al.*, 2014. Flexisp: a flexible camera image processing framework. *ACM Transactions on Graphics* 33 (6), 231.
- Jain V., Seung S.*, 2009. Natural image denoising with convolutional networks. In: *Advances in Neural Information Processing Systems*, pp. 769–776.
- Kamilov U. S., Mansour H., Wohlberg B.*, 2017. A PnP priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters* 24 (12), 1872–1876.
- Kingma D., Ba J.*, 2015. Adam: a method for stochastic optimization. In: *International Conference for Learning Representations*.
- Krull A., Buchholz T. O., Jug F.*, 2019. Noise2void-learning denoising from single noisy images. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137.
- Kruse J., Rother C., Schmidt U.*, 2017. Learning to push the limits of efficient fft-based image deconvolution. In: *IEEE International Conference on Computer Vision*, pp. 4586–4594.
- Lefkimmiatis S.*, 2017. Non-local color image denoising with convolutional neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3587–3596.
- Lehtinen J., Munkberg J., Hasselgren J., Laine S., Karras T., Aittala M., Aila T.*, 2018. Noise2noise: learning image restoration without clean data. In: *International Conference on Machine Learning*, pp. 2965–2974.
- Levin A., Weiss Y., Durand F., Freeman W. T.*, 2009. Understanding and evaluating blind deconvolution algorithms. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971.
- Li Z., Wu J.*, 2019. Learning deep cnn denoiser priors for depth image inpainting. *Applied Sciences* 9 (6), 1103.

- Lim B., Son S., Kim H., Nah S., Lee K. M.*, 2017. Enhanced deep residual networks for single image superresolution. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 136–144.
- Liu D., Wen B., Fan Y., Loy C. C., Huang T. S.*, 2018. Non-local recurrent network for image restoration. In: Advances in Neural Information Processing Systems, pp. 1673–1682.
- Ma K., Duanmu Z., Wu Q., Wang Z., Yong H., Li H., Zhang L.*, 2017. Waterloo exploration database: new challenges for image quality assessment models. IEEE Transactions on Image Processing 26 (2), 1004–1016.
- Martin D., Fowlkes C., Tal D., Malik J.*, 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE International Conference on Computer Vision, vol. 2, pp. 416–423.
- Metzler C. A., Maleki A., Baraniuk R. G.*, 2016. From denoising to compressed sensing. IEEE Transactions on Information Theory 62 (9), 5117–5144.
- Mohan S., Kadkhodaie Z., Simoncelli E. P., Fernandez-Granda C.*, 2019. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In: International Conference on Learning Representations.
- Parikh N., Boyd S. P., et al.*, 2014. Proximal algorithms. Foundations and Trends in Optimization 1 (3), 127–239.
- Plötz T., Roth S.*, 2018. Neural nearest neighbors networks. In: Advances in Neural Information Processing Systems, pp. 1087–1098.
- Portilla J., Strela V., Wainwright M. J., Simoncelli E. P.*, 2003. Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Transactions on Image Processing 12 (11), 1338–1351.
- Richardson W. H.*, 1972. Bayesian-based iterative method of image restoration. JOSA 62 (1), 55–59.
- Riegler G., Schuler S., Ruther M., Bischof H.*, 2015. Conditioned regression models for non-blind single image super-resolution. In: IEEE International Conference on Computer Vision, pp. 522–530.
- Romano Y., Elad M., Milanfar P.*, 2017. The little engine that could: regularization by denoising (RED). SIAM Journal on Imaging Sciences 10 (4), 1804–1844.
- Ronneberger O., Fischer P., Brox T.*, 2015. U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241.
- Roth S., Black M. J.*, 2009. Fields of experts. International Journal of Computer Vision 82 (2), 205–229.
- Ryu E., Liu J., Wang S., Chen X., Wang Z., Yin W.*, 2019. PnP methods provably converge with properly trained denoisers. In: International Conference on Machine Learning, pp. 5546–5557.
- Samuel K. G., Tappen M. F.*, 2009. Learning optimized MAP estimates in continuously-valued MRF models. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 477–484.
- Schmidt U., Roth S.*, 2014. Shrinkage fields for effective image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2774–2781.
- Shocher A., Cohen N., Irani M.*, 2018. «zero-shot» super-resolution using deep internal learning. In: IEEE International Conference on Computer Vision, pp. 3118–3126.

- Sun J., Tappen M. F.*, 2011. Learning non-local range Markov random field for image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2745–2752.
- Sun J., Tappen M. F.*, 2013. Separable Markov random field model and its applications in low level vision. *IEEE Transactions on Image Processing* 22 (1), 402–407.
- Sun Y., Liu J., Kamilov U.*, 2019a. Block coordinate regularization by denoising. In: *Advances in Neural Information Processing Systems*, pp. 380–390.
- Sun Y., Xu S., Li Y., Tian L., Wohlberg B., Kamilov U. S.*, 2019b. Regularized Fourier ptychography using an online PnP algorithm. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7665–7669.
- Tappen M. F.*, 2007. Utilizing variational optimization to learn Markov random fields. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.
- Teodoro A. M., Bioucas-Dias J. M., Figueiredo M. A.*, 2016. Image restoration and reconstruction using variable splitting and class-adapted image priors. In: *IEEE International Conference on Image Processing*, pp. 3518–3522.
- Timofte R., Agustsson E., Van Gool L., Yang M. H., Zhang L.*, 2017. Ntire 2017 challenge on single image superresolution: methods and results. In: *CVPRW*, pp. 114–125.
- Timofte R., Rothe R., Van Gool L.*, 2016. Seven ways to improve example-based single image super resolution. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1865–1873.
- Tirer T., Giryes R.*, 2018. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing* 28 (3), 1220–1234.
- Tirer T., Giryes R.*, 2019. Super-resolution via image-adapted denoising cnns: incorporating external and internal learning. *IEEE Signal Processing Letters* 26 (7), 1080–1084.
- Venkatakrishnan S. V., Bouman C. A., Wohlberg B.*, 2013. PnP priors for model based reconstruction. In: *IEEE Global Conference on Signal and Information Processing*, pp. 945–948.
- Venkatesh G., Naresh Y., Little S., Oonnor N. E.*, 2018. A deep residual architecture for skin lesion segmentation. In: *Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*. Springer, pp. 277–284.
- Wang X., Yu K., Wu S., Gu J., Liu Y., Dong C., Qiao Y., Loy C. C.*, 2018. ESRGAN: enhanced super-resolution generative adversarial networks. In: *The European Conference on Computer Vision Workshops*.
- Xu L., Ren J. S., Liu C., Jia J.*, 2014. Deep convolutional neural network for image deconvolution. In: *Advances in Neural Information Processing Systems*, pp. 1790–1798.
- Yair N., Michaeli T.*, 2018. Multi-scale weighted nuclear norm image restoration. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3165–3174.
- Zamir S. W., Arora A., Khan S., Hayat M., Khan F. S., Yang M. H., Shao L.*, 2020. Cycleisp: real image restoration via improved data synthesis. *IEEE Conference on Computer Vision and Pattern Recognition*.

- Zhang J., Ghanem B., 2018. ISTA-net: interpretable optimization-inspired deep network for image compressive sensing. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1828–1837.
- Zhang K., Zhou X., Zhang H., Zuo W., 2015. Revisiting single image super-resolution under Internet environment: blur kernels and reconstruction algorithms. In: Pacific Rim Conference on Multimedia, pp. 677–687.
- Zhang K., Zuo W., Chen Y., Meng D., Zhang L., 2017a. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing, 3142–3155.
- Zhang K., Zuo W., Gu S., Zhang L., 2017b. Learning deep CNN denoiser prior for image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3929–3938.
- Zhang K., Zuo W., Zhang L., 2018a. FFDNet: toward a fast and flexible solution for CNN-based image denoising. IEEE TIP 27 (9), 4608–4622.
- Zhang K., Zuo W., Zhang L., 2018b. Learning a single convolutional super-resolution network for multiple degradations. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3262–3271.
- Zhang L., Wu X., Buades A., Li X., 2011. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic Imaging 20 (2), 1–15.
- Zhang Y., Li K., Li K., Zhong B., Fu Y., 2019. Residual non-local attention networks for image restoration. In: International Conference on Learning Representations.
- Zhang Z., Liu Q., Wang Y., 2018. Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters 15 (5), 749–753.
- Zhao N., Wei Q., Basarab A., Dobigeon N., Kouamé D., Tournier J. Y., 2016. Fast single image super-resolution using a new analytical solution for 2-2 problems. IEEE Transactions on Image Processing 25 (8), 3683–3697.
- Zoran D., Weiss Y., 2011. From learning models of natural image patches to whole image restoration. In: IEEE International Conference on Computer Vision, pp. 479–486.

ОБ АВТОРАХ ГЛАВЫ

Кай Чжан получил степень доктора философии в Школе компьютерных наук и технологий Харбинского технологического института, Китай, в 2019 г. С июля 2015 г. по июль 2017 г. и с июля 2018 г. по апрель 2019 г. он работал научным сотрудником в Департаменте вычислительной техники Гонконгского политехнического университета. В настоящее время является постдокторантом в лаборатории Computer Vision Lab, Цюрих, Швейцария, где трудится с профессорами Люком Ван Гулом и профессором Раду Тимофте. Его исследовательские интересы включают машинное обучение и обработку изображений.

Раду Тимофте получил докторскую степень в области электроники в KU Leuven, Бельгия, в 2013 г. С 2013 по 2016 г. он был постдокторантом в лабо-

ратории компьютерного зрения, в Цюрихе, Швейцария. С 2016 г. был руководителем группы и преподавателем в той же лаборатории. Также является профессором и заведующим кафедрой компьютерных наук в Вюрцбургском университете, Германия, и лауреатом профессорской премии Александра фон Гумбольдта 2022 г. в области искусственного интеллекта. Член редколлегии ведущих журналов, таких как *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *Elsevier Neurocomputing*, *Elsevier Computer Vision and Image Understanding*, *SIAM Journal on Imaging Sciences*, выступал в качестве председателя на ведущих конференциях, таких как CVPR 2021, IJCAI 2021, ECCV 2020, ACCV 2020, ICCV 2019. Является соучредителем Merantix и соорганизатором мероприятий NTIRE, CLIC, AIM и PIRM. Его текущие исследовательские интересы включают глубокое обучение, неявные модели, сжатие, отслеживание, восстановление и улучшение изображений.

Глава 15

Атаки на визуальные системы и защита от злоумышленников

Авторы главы:

Чанги О, Алесслио Зомперо и Андреа Кавалларо,
Центр интеллектуального восприятия,
Лондонский университет королевы Марии,
Лондон, Соединенное Королевство

Краткое содержание главы:

- постановка проблемы состязательных атак для визуальных задач, использующих как изображения, так и видео в качестве входных данных;
- свойства состязательных атак и виды возмущений;
- описание целевых моделей и наборов данных, используемых в сценариях атаки;
- обсуждение состязательных атак на системы обработки изображений, классификации изображений, семантической сегментации, обнаружения объектов, отслеживания объектов и классификации видео;
- обсуждение средств защиты, разработанных против состязательных атак.

15.1. ВВЕДЕНИЕ

Хорошо известно, что глубокие нейронные сети (deep neural networks, DNN) успешно справляются с различными задачами компьютерного зрения, такими как классификация изображений (Krizhevsky et al., 2012; He et al., 2016), обнаружение объектов (Ren et al., 2017; Redmon, Farhadi, 2017), семантическая сегментация (Long et al., 2015; Yu et al., 2017), оценка оптического потока (Revaud et al., 2015; Ranjan, Black, 2017) и классификация видео (Carreira, Zisserman, 2017; Jiang et al., 2018; Ng et al., 2015). Однако DNN чувствительны к возмущениям входных данных, которые создают так называемые *обманные образцы* (adversarial examples), которые мы будем дальше называть *обман-*

ными изображениями, побуждающие DNN к ошибочным прогнозам (Szegedy et al., 2014).

Исследование изменений данных, предназначенных для обхода классификатора, не ново: методы, вводящие классификаторы в заблуждение, обсуждаются уже более двух десятилетий и включают атаки на системы обнаружения мошенничества (Bolton et al., 2002), спам-фильтры (Meyer, Whateley, 2004) и на конкретные классификаторы, такие как машины опорных векторов (Biggio et al., 2013). Относительно недавно возрос интерес к обманным образцам для DNN, решающих визуальные задачи (Szegedy et al., 2014).

Обманные изображения в области визуальных задач генерируются путем изменения значений пикселей с помощью тщательно созданного аддитивного шума, незаметного для человеческого глаза (Carlini, Wagner, 2017; Jiang et al., 2019); замены полукруглых или прямых областей изображения (Brown et al., 2018; Ranjan et al., 2019) либо границ изображения (Zajac et al., 2019). Обманные изображения помогают исследовать и повышать надежность моделей DNN (Tsipras et al., 2019; Allen-Zhu and Li, 2020; Engstrom et al., 2019; Santurkar et al., 2019), а также защищать личную информацию в изображениях (Li et al., 2019; Sanchez-Matilla et al., 2020).

Атака противника может быть направленной или ненаправленной. *Направленные атаки* модифицируют изображение или видео таким образом, чтобы модель DNN предсказала нужную злоумышленнику метку класса, такую как тип объекта (Szegedy et al., 2014), или нужную траекторию объекта в последующих кадрах (Liang et al., 2020). *Ненаправленные атаки* изменяют исходное изображение или видео, чтобы DNN отнесла его к любому другому классу, отличному от истинного, или сгенерировала неправильные ограничивающие рамки, чтобы ввести в заблуждение средство отслеживания (Liang et al., 2020). Наконец, тщательно измененные ключевые признаки могут привести к ложному обнаружению или неправильному обнаружению объекта другого типа (Lu et al., 2017).

После определения проблемы визуальных состязательных атак (раздел 15.2) мы обсудим их основные свойства (раздел 15.3) и типы возмущений, которые они вызывают (раздел 15.4). Затем рассмотрим сценарии атак, модели и наборы данных, используемые для создания состязательных атак (раздел 15.5). В частности, мы рассматриваем состязательные атаки для задач обработки изображений (раздел 15.6), классификации изображений (раздел 15.7), семантической сегментации и обнаружения объектов (раздел 15.8), отслеживания объектов (раздел 15.9) и классификации видео (раздел 15.10). Наконец, мы представляем стратегии защиты моделей DNN от этих атак (раздел 15.11) и завершаем главу (раздел 15.12).

15.2. ОПРЕДЕЛЕНИЕ ПРОБЛЕМЫ

Пусть x – изображение или видео, а y – истинная метка, связанная с x . Метка может быть отдельным классом для классификации изображения или видео, или областью оптического потока для обнаружения либо отслеживания объекта.

Пусть $f(\cdot)$ – модель DNN, отображающая \mathbf{x} в метку y : $y = f(\mathbf{x})$. Атака модифицирует \mathbf{x} , чтобы создать обманное изображение $\hat{\mathbf{x}}$, которое вводит работающую модель в заблуждение, так что $f(\mathbf{x}) = f(\hat{\mathbf{x}})$ (рис. 15.1). Обманное изображение можно получить, напрямую изменив визуальные данные с возмущением δ таким, что $\hat{\mathbf{x}} = g(\mathbf{x}, \delta)$, где $g(\cdot)$ представляет собой процесс добавления возмущения к \mathbf{x} или замены значений пикселей в \mathbf{x} на взятые из δ .

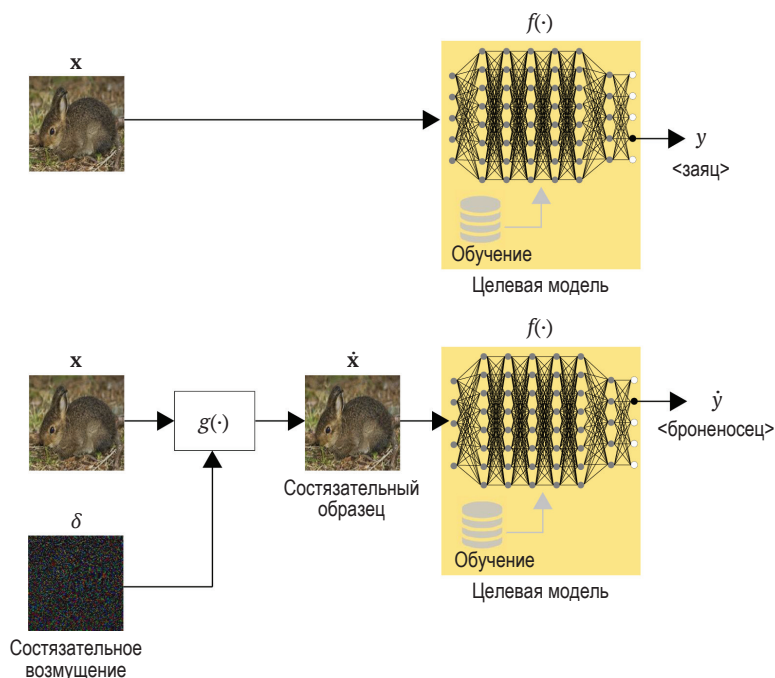


Рис. 15.1 ❖ Наглядный пример состязательной атаки на классификатор изображений. Классификатор атакуемой модели присваивает исходному изображению метку «заяц» (вверху), а обманному изображению, полученному с помощью атаки базового итеративного метода (Kurakin et al., 2017), та же целевая модель присваивает метку «броненосец» (внизу). Целевая модель представляет собой иллюстративное и абстрактное представление классификатора Inception V3 (Szegedy et al., 2016), обученного в ImageNet (Deng et al., 2009). Стоит отметить, что в целях наглядности показанное здесь направленное возмущение в 20 раз больше, чем реальное

Обманное возмущение может быть создано с помощью машинного обучения. В таком случае злоумышленник может разработать функцию потерь $\mathcal{L}(\mathbf{x}, y)$, которая используется для генерации прогноза входного сигнала относительно правильного прогноза (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Jiang et al., 2019; Shamsabadi et al., 2020c; Liang et al., 2020). Например, потеря с точки зрения атакующего может *максимизировать* ошибку классификации для набора обучающих изображений (Szegedy et al., 2014; Moosavi-Dezfooli et al., 2016; Hosseini, Poovendran, 2018) или может опти-

мизировать функцию потерь с помощью дополнительной цели изменения изображения с незаметным возмущением (Szegedy et al., 2014; Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Modas et al., 2019; Shamsabadi et al., 2020).

Основываясь на информации о целевой модели $f(\cdot)$ и/или ее прогнозах u , доступных злоумышленнику, атаку можно классифицировать как *белый ящик* или *черный ящик*. При атаке белого ящика злоумышленник имеет полный доступ к целевой модели (конкретной архитектуре, ее параметрам и/или обучающим данным) и пользуется им для непосредственного создания возмущения. Параметры белого ящика помогают обнаружить ограничения обученной модели и оценить их надежность. При атаке методом черного ящика злоумышленник не имеет прямого доступа к архитектуре или параметрам целевой модели. Атаки методом черного ящика более реалистичны, поскольку модели (или даже их выходные данные) чаще всего недоступны в реальных ситуациях. При атаке методом черного ящика злоумышленник может не иметь доступа к выходным данным $f(\cdot)$ и, следовательно, никакой информации об u (*атака без вывода*), или иметь доступ только к метке u (классу), сгенерированной обманными образцами, поданными в классификатор $f(\cdot)$ (*атака с выводом метки*), или только к значениям нейронов до последнего слоя, а именно к логитам/вероятностям обманных образцов, представленных классификатору (*атака с распределением-выводом*).

15.3. Свойства состязательной атаки

Атаки со стороны противника можно оценить на основе четырех основных характеристик, а именно эффективности, робастности, переносимости и заметности.

Эффективность состязательной атаки – это степень, в которой ей удастся ввести в заблуждение модель машинного обучения. Эффективность можно измерить как точность модели $f(\cdot)$ на целевом наборе данных. Чем ниже точность, тем выше эффективность атаки противника.

Робастность (устойчивость) состязательной атаки – это ее эффективность при наличии защиты, которая устраняет влияние δ до того, как данные будут обработаны моделью $f(\cdot)$. Примеры защиты включают медианную фильтрацию (Xu et al., 2018), повторное квантование (Xu et al., 2018) и сжатие JPEG (Das et al., 2017; Dziugaite et al., 2016; Guo et al., 2018). Робастность может быть измерена как разница в точности $f(\cdot)$ на целевом наборе данных, когда защита используется по отношению к ситуации, когда защита не используется. Чем меньше эта разница, тем выше устойчивость атаки.

Переносимость атаки противника – это степень, в которой возмущение δ , созданное для модели $f(\cdot)$, эффективно для введения в заблуждение другой модели $f'(\cdot)$, не применявшейся для создания δ . Переносимость может быть измерена как разница в точности $f(\cdot)$ и $f'(\cdot)$ на целевом наборе данных обманных образцов, созданных для $f(\cdot)$. Чем меньше эта разница, тем выше переносимость атаки на классификатор $f'(\cdot)$.

Заметность состязательной атаки – это степень, в которой злонамеренное возмущение δ может быть замечено как таковое человеком, смотрящим на изображение или видео. Заметность можно измерить с помощью теста двойного воздействия, в котором сравниваются пары изображений или видеоклипов $\{(x, \hat{x})\}$; однократного теста на естественность изображения или видеофрагмента $\{\hat{x}\}$, проверяемого по результатам, полученным с соответствующим $\{x\}$; или с (надежной) мерой качества восприятия без эталона.

В дополнение к этим четырем основным свойствам при анализе или оценке состязательных атак для конкретных задач или целей могут учитываться и другие свойства, такие как обнаруживаемость и обратимость. *Обнаруживаемость* атаки со стороны – это степень, в которой защитный механизм способен заметить, что возмущение было преднамеренно применено для изменения исходного изображения, видео или сцены. *Обнаруживаемость*, которая связана с робастностью, может быть измерена как доля обманных изображений, которые обнаруживаются как таковые в заданном наборе данных или заданных сценариях. *Обнаруживаемость* атаки можно оценить путем сравнения выходных данных $f(\cdot)$ на заданных входных данных и на тех же входных данных, предварительно обработанных защитой, поскольку разные выходные данные предполагают наличие атаки. Наконец, *обратимость* состязательной атаки – это степень, в которой анализ предсказаний или выходных меток $f(\cdot)$ позволяет извлечь исходный класс \hat{x} . Например, анализ частоты сопоставления состязательного и исходного предсказаний показал, что нецелевые атаки более обратимы, чем целевые (Li et al., 2021).

15.4. Типы возмущений

Злонамеренное возмущение может быть глобальным или локальным в зависимости от пространственного распределения δ , а также ограниченным или неограниченным в зависимости от количества изменений, вызванных интенсивностью пикселей.

Глобальные возмущения изменяют интенсивность каждого пикселя в отдельности и, таким образом, генерируют сильный пространственно-частотный шум. Природа этих возмущений делает их уязвимыми для средств защиты (например, медианной фильтрации или сжатия JPEG) (Dziugaite et al., 2016). Глобальные возмущения могут быть разреженными (Papernot et al., 2016b) или плотными (Goodfellow et al., 2015). Крайним случаем является однопиксельное возмущение, которое изменяет один пиксель исходного изображения (Su et al., 2019).

Возмущения области применяются к областям изображения, таким как рамка вокруг границы изображения, семантическая область или прямоугольный или круглый патч. Злонамеренные патчи чрезвычайно заметны для целевой модели $f(\cdot)$ (Brown et al., 2018; Ranjan et al., 2019). Выбранными областями также можно манипулировать в зависимости от их ожидаемой значимости для зрительной системы человека (Shamsabadi et al., 2020c).

Ограниченные возмущения лимитируют величину изменения значения каждого пикселя (Goodfellow et al., 2015) или изображения (Szegedy et al., 2014), как правило, с ограничением ℓ_p -нормы. Граница возмущения может быть определена параметром ϵ таким, что $\|\mathbf{x} - \hat{\mathbf{x}}\|_p < \epsilon$ (Carlini, Wagner, 2017; Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Modas et al., 2019). Использование нормы ℓ_2 ($\|\cdot\|_2$) ограничивает максимальное изменение энергии (Moosavi-Dezfooli et al., 2016), тогда как использование нормы ℓ_∞ ($\|\cdot\|_\infty$) ограничивает максимальное изменение для каждого пикселя (Kurakin et al., 2017; Li et al., 2019). Ограниченные возмущения, как правило, имеют ограниченную переносимость (Xie et al., 2019) и ограниченную устойчивость к средствам защиты (Xu et al., 2018).

Неограниченные возмущения не ограничивают изменения интенсивности или области атаки, что часто приводит к заметно искаженным обманным образцам (Hosseini, Poovendran, 2018). Чтобы уменьшить (или избежать) артефакты, неограниченные возмущения на основе контента манипулируют низкоуровневыми признаками изображения, такими как цвет, текстура или края, и обеспечивают более высокую переносимость и надежность (Bhattad et al., 2020; Shamsabadi et al., 2020b).

15.5. СЦЕНАРИИ АТАКИ

Мы сгруппировали модели, на которые нацелены атаки злоумышленников в задачах обработки изображений и компьютерного зрения, а также соответствующие наборы данных, используемые для создания и оценки этих атак на изображения и видео. В табл. 15.1 и 15.2 приведены основные характеристики состязательных атак на изображения или видео.

15.5.1. Целевые модели

Целевая модель DNN имеет определенную архитектуру, состоящую из слоев, параметры которых изучаются. Количество слоев определяет глубину модели DNN. Увеличение глубины архитектуры DNN приводит к увеличению количества параметров, и, следовательно, для изучения значений параметров во время обучения требуется крупномасштабный набор данных (более миллиона изображений). Последние слои могут предсказывать logits, т. е. значения в диапазоне $(-\infty, +\infty)$, или вероятности (значения в диапазоне $[0, 1]$) предсказанных меток либо конечных меток. Поскольку не все слои (например, слои пулинга) имеют обучаемые параметры, в этой главе мы будем использовать термин «слой» только для тех из них, у которых есть обучаемые параметры (например, сверточные и полносвязные слои).

Атакованными (целевыми) моделями являются LeNet, AlexNet, VGGNets, GoogleNet и варианты ResNets и WideResNets, DenseNets и MobileNets для классификации изображений; FCN, HED и Faster R-CNN для семантической сегментации, обнаружения границ и обнаружения объектов соответствен-

Таблица 15.1. Сводка существующих составительных атак для задач, использующих изображения в качестве входных данных. Обозначения: ○ – белый ящик, ● – черный ящик, T – целевая, T̄ – нецелевая, B – нецелевая, B̄ – нецелевая, Opt – неограниченная, Opt – на основе оптимизации, Grad – на основе градиента, Bound – граничное приближение, GradE – градиентная оценка, LocS – локальный поиск, RapS – случайный поиск, C – классификация, S – семантическая сегментация, D – обнаружение объектов, IC – подписи к изображениям, E – обнаружение границы

Источник	Метод	Ящик	T	T̄	B	B̄	Подход	Набор данных	Задача
(Szegedy et al., 2014)	L-BFGS	○	✓	✓	✓		Opt	ImageNet, MNIST, Youtube	C
(Carlini, Wagner, 2017)	CW	○	✓	✓	✓		Opt	MNIST, CIFAR	C
(Goodfellow et al., 2015)	FGSM	○	✓	✓	✓		Grad	MNIST	C
(Kurakin et al., 2017b)	BIM (I-FGSM)	○	✓	✓	✓		Grad	ImageNet	C
(Madry et al., 2018)	PGD	○	✓	✓	✓		Grad	MNIST, CIFAR	C
(Papernot et al., 2016b)	JSMA	○	✓	✓	✓		Grad	MNIST	C
(Moosavi-Dezfooli et al., 2016)	DeepFool	○		✓	✓		Grad	ImageNet, MNIST, CIFAR	C
(Modas et al., 2019)	SparseFool	○		✓	✓		Grad	ImageNet, MNIST, CIFAR	C
(Xie et al., 2019)	di ² -fgsm	○, ●	✓	✓	✓		Grad	ImageNet	C
(Tramer et al., 2018)	E-FGSM	○, ●	✓		✓		Grad	ImageNet	C
(Li et al., 2019a)	P-FGSM	○	✓		✓		Grad	Places	C
(Sanchez-Matilla et al., 2020)	RP-FGSM	○	✓	✓	✓		Grad	Places	C
(Moosavi-Dezfooli et al., 2017)	DAP	○		✓	✓		Opt	ImageNet	C
(Mopuri et al., 2017)	Быстрый обман признаков	○		✓	✓		Opt	ImageNet, Places-205	C
(Baluja, Fischer, 2018)	ATN	○	✓	✓	✓		Opt	ImageNet, MNIST	C
(Xiao et al., 2018)	AdvGAN	○, ●	✓	✓	✓		Opt	ImageNet, MNIST	C
(Poursaeed et al., 2018)	GAP	○	✓	✓	✓		Opt	ImageNet, Cityscapes	C, S
(Mopuri et al., 2018)	NAG	○, ●		✓	✓		Opt	ImageNet	C
(Bhattad et al., 2020)	Семантическая манипуляция	○	✓			✓	Opt	ImageNet, MSCOCO	C, IC
(Shamsabadi et al., 2020a)	EdgeFool	○		✓		✓	Opt	ImageNet, Places	C

(Papemot et al., 2016a)	SBA	●	✓	✓	Bound	MNIST	C
(Shi et al., 2019)	Curls & Whey	●	✓	✓	Bound	ImageNet	C
(Dong et al., 2019)	Атака TI	●	✓	✓	Bound	ImageNet	C
(Chen et al., 2017)	ZOO	●	✓	✓	GradE	ImageNet, MNIST, CIFAR	C
(Плюс et al., 2018)	Атака с ограничением запроса	●	✓	✓	GradE	ImageNet	C
(Tu et al., 2019)	AutoZOOM	●	✓	✓	GradE	ImageNet, MNIST, CIFAR	C
(Narodytska, KasiViswanathan, 2017)	LocSearchAdv	●	✓	✓	GradE	ImageNet, MNIST, CIFAR, SVHN, STL	C
(Brendel et al., 2018)	BA	●	✓	✓	LocS	ImageNet, MNIST, CIFAR	C
(Guo et al., 2019)	SimBA	●	✓	✓	LocS	ImageNet	C
(Hosseini, Poovendran, 2018)	SemanticAdv	●	✓	✓	RanS	CIFAR	C
(Shamsabadi et al., 2020c)	ColorFool	●	✓	✓	RanS	ImageNet, CIFAR, Places	C
(Fischer et al., 2017)	SSA	○	✓	✓	Grad	Cityscapes	S
(Xie et al., 2017)	DAG	○	✓	✓	Grad	voc	S,D
(Wei et al., 2019)	UEA	○, ●	✓	✓	Opt	ImageNet VID	D
(Cosgrove, Yuille, 2020)	Граничная атака	○	✓	✓	Grad	Cityscapes	E

Таблица 15.2. Сводка существующих состязательных атак для задач, использующих видео в качестве входных данных.
 Обозначения: ○ – белый ящик, ● – черный ящик, Т – целевая, Т – нецелевая, DD – зависимость ввода данных, U – универсальная, В – неограниченный, В – ограниченная, R – региональная, Gen – генеративная, Opt – оптимизация, С – классификация видео, ME – оценка движения, OT – отслеживание объектов

Источник	Метод	Ящик	Т	Т	DD	U	В	В	R	Подход	Набор данных	Задача
(Ranjan et al., 2019)	OFA	○, ●	✓	✓	✓	✓	✓	✓	✓	Opt	KITTI	ME
(Liang et al., 2020)	FAN	○, ●	✓	✓	✓	✓	✓	✓	✓	Gen	OTB, VOT	OT
(Jiang et al., 2018)	V-BAD	●	✓	✓	✓	✓	✓	✓	✓	Opt	DCF, HMDB, Kinetics	C
(Li et al., 2019)	C-DUP	○	✓	✓	✓	✓	✓	✓	✓	Gen	UCF, JESTER	C
(Lo, Patel, 2020)	MultAV	○	✓	✓	✓	✓	✓	✓	✓	Direct, Opt	UCF	C

но; C3D, CNN+LSTM, I3D и 3D ResNet-18 для классификации видео; SiamFC, SiamRPN, SiamRPN+CIR и SiamRPN++ для визуального отслеживания объектов; FlowNet, FlowNet2, SpyNet, PWC-Net и Back2Future для оценки движения. Мы также включили две модели, отличные от DNN, а именно Epic Flow и LDOF, для оценки оптического потока.

15.5.1.1. Модели для задач, связанных с изображениями

LeNet – это сверточная нейронная сеть (CNN) с 3 слоями свертки и 2 полносвязными слоями для распознавания изображений (LeCun et al., 1998).

AlexNet имеет 8 сверточных слоев и 3 полносвязных слоя, а архитектура модели аналогична LeNet (Krizhevsky et al., 2012).

VGGNet имеет более глубокую архитектуру с 16 и 19 уровнями, но ядра сверточных фильтров меньшего размера, поскольку несколько небольших сверточных фильтров с меньшим количеством параметров превосходят один большой фильтр (Simonyan, Zisserman, 2014).

GoogLeNet (Inception-v1) имеет 22 слоя (учитываются только слои с параметрами), и 9 из этих слоев состоят из сверточных фильтров нескольких размеров (начальный слой/модуль), что снижает вычислительные затраты и ресурсы архитектуры (Szegedy et al., 2015). Inception-v2 и Inception-v3 (Szegedy et al., 2016) и Inception-v4 (Szegedy et al., 2017) – это варианты, которые дополнительно повышают точность и снижают вычислительную сложность. Улучшения включают разложение фильтров на стек фильтров меньшего размера, увеличение количества фильтров и их размера в каждой модели (шире, а не глубже), сглаживание меток (член регуляризации для снижения доверия сети к классу), добавление блоков редукции и упрощение различных модулей (например, за счет выбора разного количества фильтров и их размера). Блоки редукции представляют собой сверточные слои 1×1 , также известные как проекционные слои, которые выполняют объединение карт признаков по измерению канала, что приводит к уменьшению размерности.

ResNet может иметь 18, 50, 101 или 150 слоев. Блоки слоев сгруппированы для изучения функции, которая сопоставляет входные данные с выходными в виде остаточного отображения (He et al., 2016).

DenseNet имеет архитектуры со 121, 169, 201 и 264 слоями, чьи плотные блоки могут содержать от 12 до 128 сверточных слоев. DenseNet расширяет остаточное отображение ResNet, предоставляя карты признаков всех предыдущих слоев в блоке в качестве входных данных для последующих слоев в том же блоке (плотное соединение), чтобы уменьшить количество параметров, улучшить поток информации и градиенты по всей сети и уменьшить переобучение (Huang et al., 2017).

MobileNet имеет 28 уровней и два гиперпараметра, которые регулируют ширину сети на каждом уровне и разрешение входного изображения (Howard et al., 2017). Первые 13 слоев представляют собой разделимые по глубине слои свертки, которые разбивают стандартный слой свертки на два слоя: отдельный слой для фильтрации и отдельный слой для объединения результирующего вывода. Эта факторизация уменьшает объем вычислений и размер модели. Архитектуру можно дополнительно уменьшить, удалив слои.

WideResNet имеет более высокую точность, чем ResNet, в задачах классификации изображений с более мелкой архитектурой и большей шириной остаточных блоков (Zagoruyko, Komodakis, 2016).

Полностью сверточные сети (fully convolutional networks, FCN) используют модели CNN, такие как AlexNet, VGGNet и GoogleNet, для классификации изображений, но заменяют полносвязные слои сверточными слоями, за которыми следует слой повышающей дискретизации (Long et al., 2015), таким образом создавая выходную карту признаков того же размера, что и входная. Из-за этого FCN можно обучить сквозной (семантической) сегментации.

Целостно-вложенный детектор краев (holistically-nested edge detector, HED) – это архитектура на основе FCN, которая использует средний пул для обнаружения краев (Xie, Tu, 2015).

Faster R-CNN – это архитектура на основе FCN, которая учится генерировать объекты-кандидаты с последующей локализацией и классификацией объектов среди кандидатов (Ren et al., 2017). Faster R-CNN состоит из предварительно обученных слоев свертки, таких как 13 слоев свертки VGGNet, где выходные данные подаются в небольшую сеть с двумя слоями свертки, которая выводит признаки более низкой размерности. Эти признаки, каждый из которых представляет объект-кандидат, затем передаются в два независимых полносвязных слоя, чтобы найти расположение ограничивающих рамок и классы для объектов-кандидатов.

15.5.1.2. Модели для видеозадач

CNN+LSTM – это классификатор видео, который применяет 2D-архитектуры (например, AlexNet или GoogleNet) к каждому кадру независимо, используя общие параметры во времени, а затем учится интегрировать информацию во времени с помощью рекуррентной нейронной сети, основанной на LSTM, работающей на активациях CNN на уровне кадра (Ng et al., 2015). Комбинация LSTM с 2D-сетями позволяет классификатору видео поддерживать постоянное количество параметров при извлечении глобального описания временной эволюции видео. **C3D** – это пространственно-временной классификатор с 8 трехмерными сверточными слоями и 2 полносвязными слоями (Tran et al., 2015). C3D может обрабатывать 16 кадров для распознавания действий.

Inflated 3D CNN (I3D) – это пространственно-временная архитектура, построенная на основе 2D DNN для классификации изображений (например, InceptionV1), которая объединяет выходные данные двух 3D CNN, одна из которых обрабатывает группу кадров RGB, а другая – группу предсказаний оптического потока среди последовательных кадров RGB (Carreira, Zisserman, 2017). I3D расширяет (inflate) фильтры и операции объединения с 2D на 3D. Модель I3D также может использовать предварительно обученные веса из 2D-моделей в качестве инициализации перед точной настройкой.

3D ResNet – это пространственно-временная архитектура, которая, подобно I3D, расширяет модели ResNet на основе изображений во временную область с использованием 3D CNN (Hara et al., 2018). Дополнительные модели, такие как WideResNet и DenseNet, могут быть расширены с помощью 3D CNN.

SiamFC использует полностью сверточную сиамскую архитектуру, состоящую из двух сверточных нейронных сетей с одинаковыми параметрами. SiamFC обучен прогнозировать карту сходства между целевым шаблоном (или эталонным патчем) и областью поиска в текущем изображении посредством взаимной корреляции функций, выдаваемых двумя ветвями. Эталонный патч предоставляется или инициализируется в первом кадре (Bertinetto et al., 2016).

SiamRPN основана на сиамской архитектуре для извлечения признаков из шаблона и области поиска. В SiamRPN добавлена сеть прогнозов по регионам, состоящая из двух ветвей, одной для классификации переднего и заднего планов, а другой для регрессии, основанной на парной взаимной корреляции (Li et al., 2018).

SiamRPN++ – это улучшенная версия SiamRPN, использующая стратегию выборки с учетом пространства, которая обеспечивает строгую трансляционную инвариантность, поскольку заполнение, введенное в DNN, нарушает инвариантность. Кроме того, в SiamRPN++ добавлен уровень взаимной корреляции по глубине, который прогнозирует многоканальные признаки корреляции между шаблоном и патчами поиска, используя структуру ResNet (Li et al., 2019).

SiamRPN+CIR добавляет блоки Cropping-Inside Residual (CIR), чтобы устранить базовое смещение положения, вызванное заполнением нулями. SiamRPN+CIR применяет блоки CIR к различным глубоким и широким сетям, таким как ResNet и Inception, при использовании в SiamFC и SiamRPN (Zhang, Peng, 2019).

FlowNet – это автокодировщик с 9 сверточными слоями и корреляционным слоем, который выполняет мультипликативное сравнение патчей между двумя картами признаков. Пары изображений произвольного размера предоставляются в качестве входных данных для двух ветвей архитектуры и объединяются со слоем корреляции. Часть декодера имеет 4 слоя обратной свертки, которые выполняют расширение карт признаков и объединение с картами признаков из кодировщика для уточнения деталей изображения оптического потока (Dosovitskiy et al., 2015).

FlowNet2 объединяет несколько архитектур FlowNet для повышения точности оценки оптического потока как при малых, так и при больших смещениях. Первой сетью является FlowNet, а в стековых сетях используется одна ветвь, вход которой определяется конкатенацией потока, сформированного предыдущей сетью с двумя изображениями – искаженным изображением, основанным на оптическом потоке, и ошибкой яркости. Последний этап FlowNet2 объединяет выходные данные стековых сетей с выходными данными автокодировщика для получения окончательного потока (Ilg et al., 2017).

SpyNet (spatial pyramid network, сеть пространственных пирамид) сочетает в себе стратегию «от грубого к точному», основанную на формуле пространственной пирамиды, со сверточными нейронными сетями для оценки больших движений внутри пирамиды изображения. Для каждого уровня пирамиды сверточная нейронная сеть обновляет вычисленный оптический поток, поскольку одно изображение пары переносится вычисленным оптическим потоком на более грубом уровне (Ranjan, Black, 2017).

PWC-Net также представляет собой архитектуру «от грубого к точному», которая заменяет пирамиду изображений пирамидой признаков. На каждом уровне пирамиды PWC-Net добавляет слой, который переносит признаки от второго изображения к первому изображению, используя поток с повышенной дискретизацией, и слой корреляции между признаками первого и второго изображений для вычисления стоимости ассоциирования пикселя с соответствующими пикселями в следующем кадре (Sun et al., 2018).

Back2Future – это архитектура «от грубого к точному», которая использует как изображения, так и пирамиды признаков из трех последовательных кадров (прошлого, настоящего и будущего) и обучается без учителя с фотометрическими потерями. Формула мультикадра учитывает окклюзии и позволяет добавить модель линейного движения в качестве мягкого временного ограничения (Janai et al., 2018).

Epic Flow – это *классический* подход к вычислению оптического потока, который выполняет плотное сопоставление путем интерполяции с сохранением границ из разреженного набора совпадений с последующей минимизацией вариационной энергии, инициализируемой плотными совпадениями. Интерполяция «от разреженного к плотному» основывается на геодезическом расстоянии с учетом границ, адаптированном для обработки окклюзий и границ движения (Revaud et al., 2015).

LDOF – это *классический* подход к вычислению оптического потока, который сочетает в себе сопоставление локальных дескрипторов (то есть векторов признаков, извлеченных из локальной области вокруг локализованных точек внимания, таких как угловые точки) с вариационными методами, основанными на переносе изображения и минимизации энергии. Этот подход может обрабатывать быстрые движения различных частей тела (Brox, Malik, 2011).

15.5.2. Наборы данных и метки

Далее мы опишем основные характеристики наборов данных (и их аннотаций), используемых для создания злонамеренных изображений и видео. К наборам данных изображений относятся MNIST, SVHN, CIFAR-10, STL-10, ImageNet, Places, COCO. В наборы видеоданных входят наборы данных распознавания действий, используемые для классификации видео, отслеживания визуальных объектов и оценки движения. Это наборы данных KITTI, UCF, JESTER, HMDB, Kinetics и два бенчмарка для отслеживания (и их варианты): Object Tracking Benchmark (OTB) (Wu et al., 2015) и Visual Object Tracking (VOT).

15.5.2.1. Наборы данных изображений

MNIST (Modified National Institute of Standards and Technology) содержит 60 000 обучающих и 10 000 тестовых изображений (28×28, оттенки серого) рукописных цифр (LeCun, 1998).

SVHN (Street View House Numbers) содержит 600 000 изображений Google Street View с небольшими обрезанными цифрами (Netzer et al., 2011).

CIFAR-10 (Canadian Institute for Advanced Research-10) имеет 50 000 обучающих и 10 000 тестовых изображений 32×32 , разбитых на 10 классов, таких как самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль, грузовик (Krizhevsky et al., 2009).

STL-10 содержит 500 обучающих изображений, по 800 тестовых изображений на класс, и 100 000 немаркированных изображений в рамках тех же 10 классов, что и CIFAR-10. Цветовое разрешение изображения составляет 96 цветов (Coates et al., 2011).

ImageNet содержит 14 197 122 изображения RGB, разбитых на 1000 классов (Deng et al., 2009). Существует миниатюрная версия ImageNet, которая содержит 120 000 изображений из 200 классов с размером 64.

Places имеет две ориентированные на сцены версии: Places-205, которая содержит 2 448 873 изображения и 205 категорий сцен, и Places-365, в которой 1 800 000 изображений и 365 категорий сцен (Zhou et al., 2017).

COCO (Common Objects In Context) содержит 330 000 изображений, из которых более 200 000 семантически помечены для обнаружения объектов, семантической сегментации, классификации изображений и оценки ключевых точек (Lin et al., 2014).

F-MNIST (Fashion MNIST) содержит 60 000 обучающих и 10 000 тестовых изображений глубиной 28 градаций серого с меткой для 10 видов одежды, таких как футболка, брюки, пуловер, платье, пальто, сандалии, рубашка, кроссовки, сумка и высокий ботинок (Xiao et al., 2017).

VOC (PascalVOC) использовался для нескольких тестов классификации изображений, обнаружения объектов и семантической сегментации, таких как VOC2007 и VOC2012 (Everingham et al., 2015). VOC2007 состоит из 9963 изображений с 24 640 аннотированными объектами из 20 классов. VOC2012 состоит из 11 530 обучающих/проверочных изображений, содержащих 27 450 аннотированных объектов (ограничивающий прямоугольник) и 6929 сегментов из 20 классов.

LFW (Labeled Faces in the Wild) состоит из более чем 13 000 фотографий лиц, помеченных именем изображенного человека (Huang et al., 2008). Набор данных содержит одну или несколько отдельных фотографий для 1680 человек.

BSDS500 содержит 500 естественных изображений с аннотациями, нарисованными людьми для определения границ (Martin et al., 2004).

15.5.2.2. Наборы видеоданных

Cityscapes состоит из стереоскопических видеопоследовательностей с улиц 50 городов и включает 5000 изображений с аннотациями высокого качества на уровне пикселей и 20 000 дополнительных изображений с грубыми аннотациями (Cordts et al., 2016).

HAR (Human Activity Recognition, распознавание деятельности человека) содержит 30 субъектов, выполняющих повседневную деятельность с закрепленным на талии смартфоном со встроенными инерциальными датчиками (Anguita et al., 2013). Каждому участнику было предложено дважды выполнить разработанный протокол действий, включающий 6 действий, а именно

ходьбу, подъем по лестнице, спуск по лестнице, сидение, стояние и лежание. Выполнение протокола занимает в общей сложности 192 секунды.

UAV содержит 123 последовательности (всего 110 000 кадров), снятые с беспилотных летательных аппаратов на малой высоте для отслеживания объектов (Mueller et al., 2016).

KITTI состоит из 389 сцен (пар изображений), разделенных на наборы для обучения и тестирования, статической среды, снятой камерой, установленной на движущейся машине (Geiger et al., 2012). Учебный набор включает в себя разреженные аннотации потока между двумя изображениями.

UCF содержит 13 320 роликов на YouTube из 101 грубой категории человеческих действий, разделенных на 25 групп (по 4–7 клипов в каждой группе) и имеющих общие атрибуты, такие как одинаковый фон или действующие лица (Soomro et al., 2012). В наборе данных есть различные движения камеры, масштабы объектов, условия освещения, фоны и действующие лица.

JESTER содержит 148 092 видеоклипа (в среднем продолжительностью 3 секунды) с детализированными человеческими жестами, разделенными на 27 категорий и исполненными в общей сложности 1376 актерами. Примеры жестов: увеличение двумя пальцами, вытягивание руки и смахивание вправо (Materzynska et al., 2019).

HMDB содержит 6766 видеоклипов различных действий, разделенных на 51 категорию (около 100 клипов в каждой категории) и 5 типов, таких как общие мимические действия (например, улыбка), мимические действия с манипулированием предметами (например, курение), общие движения тела (например, стойка на руках), движения тела при взаимодействии с объектом (например, обливание) и движения тела для взаимодействия с человеком (например, фехтование) (Kuehne et al., 2011).

Kinetics содержит 306 245 клипов с 400 человеческими действиями, включая действия одного человека, взаимодействия человек–человек и человек–объект, изображающие различных действующих лиц и большие вариации фона, освещения, перспективы и выполнения действий (например, скорость и позы). (Кей и др., 2017). Каждая категория действий содержит от 400 до 1000 видеоклипов, продолжительность которых составляет около 10 с.

OTV (или OTV-2015) содержит 100 коротких видеороликов с покадровыми аннотациями ограничивающих рамок и 11 атрибутов, таких как изменение освещения и масштаба, окклюзия, деформация нежесткого объекта или отсутствие его в поле зрения (Wu et al., 2015). OTV-13, первый выпуск, содержал 51 последовательность.

VOT¹ содержит 60 видеороликов из менее чем 1500 кадров со скоростью 30 кадров в секунду с различными животными, людьми или движущимися объектами, часто снятыми с помощью движущейся камеры. Примеры сцен включают человека, несущего книгу, гимнастику, движущихся муравьев, летающих птиц, движущиеся дроны, едущую по дороге машину, рыбу в резервуаре с водой, людей, играющих в баскетбол, футбол и гандбол. Доступны покадровые аннотации повернутых ограничивающих рамок, масок сегментации и атрибутов.

¹ <https://www.votchallenge.net>.

15.6. ОБРАБОТКА ИЗОБРАЖЕНИЙ

В этом разделе мы обсудим состязательные атаки, направленные на вводящие в заблуждение операции обработки изображений, такие как обнаружение границ (Cosgrove, Yuille, 2020) и оценка движения (Ranjan et al., 2019).

Граничная атака (edge attack) создает злонамеренные возмущения для модели обнаружения границ на основе CNN – HED (Xie and Tu, 2015). HED обучается на наборе BSDS500 и объединяет карты признаков, извлеченные из каждого слоя свертки, чтобы классифицировать, принадлежит каждый пиксель изображения границе объекта или нет. Граничная атака вычисляет функцию потерь для обнаружения краев и создает возмущения, которые максимизируют градиент функции потерь (Goodfellow et al., 2015).

Атака оптического потока (optical flow attack, OFA) (Ranjan et al., 2019) направлена на то, чтобы нарушить две последовательные пары изображений $(\mathbf{x}_k, \mathbf{x}_{k+1})$ в видео таким образом, чтобы метки, связанные с оптическим потоком, т. е. чтобы векторы смещения (u, v) , были предсказаны неправильно (нецелевая атака). Из-за ограниченной доступности наборов данных с плотными аннотациями оптического потока OFA использует прогнозирование оптического потока модели $f(\cdot)$ в качестве псевдоаннотации для самоконтроля изучения обманного возмущения δ на основе *области* (например, круглого пятна). OFA применяет (например, вставляет на изображение) изученный патч. В дополнение к возмущению небольшого числа пикселей в паре последовательных изображений OFA ограничивает возмущение квазинезамечностью, так что

$$\|\mathbf{x}_k - \dot{\mathbf{x}}_k\|_0 + \|\mathbf{x}_{k+1} - \dot{\mathbf{x}}_{k+1}\|_0 < \epsilon, \quad (15.1)$$

где ϵ – малая постоянная (ограниченное возмущение), в то время как прогнозируемый выход модели оптического потока $f(\cdot, \cdot)$ на возмущенных изображениях существенно меняется, т. е.

$$\|f(\mathbf{x}_k, \mathbf{x}_{k+1}) - f(\dot{\mathbf{x}}_k, \dot{\mathbf{x}}_{k+1})\| > E, \quad (15.2)$$

где E – большая константа.

Таким образом, обманный патч изучается как

$$\delta = \underset{\delta}{\operatorname{argmin}} \mathbb{E}_{(\mathbf{x}_k, \mathbf{x}_{k+1}) \sim \mathcal{X}, l \sim \Omega, t \sim \mathcal{T}} \left[\frac{(u, v) \cdot (\dot{u}, \dot{v})}{\|(u, v)\| \cdot \|(\dot{u}, \dot{v})\|} \right] \quad (15.3)$$

с

$$(u, v) = f(\mathbf{x}_k, \mathbf{x}_{k+1}), \quad (15.4)$$

$$(\dot{u}, \dot{v}) = f(g(\mathbf{x}_k, \delta, l, t(\delta)), g(\mathbf{x}_{k+1}, \delta, l, t(\delta))), \quad (15.5)$$

где δ – обманный патч, l – местоположение пикселя, выбранное из всех местоположений Ω на изображении, $g(\cdot)$ – оператор, заменяющий пиксели изображения значениями δ в местоположении l , $t(\delta)$ – двумерное преобразование, выбранное из набора преобразований \mathcal{T} (или их комбинации)

и примененное к обманному патчу δ , а (\dot{u}, \dot{v}) – оптический поток, возникающий в результате атак изображений с помощью обманного патча.

OFA работает с различными типами моделей оптических потоков, такими как классические (Epic Flow, LDOF), архитектуры на основе автоэнкодера (FlowNet, FlowNet2) и архитектуры на основе изображений-пирамид (SpyNet, PWC-Net, Back2Future) – в сценарии «белого ящика». В таком сценарии OFA изучает конкретный злонамеренный патч для каждой модели независимо и разных размеров, в то время как универсальный злонамеренный патч извлекается из некоторых моделей и применяется к входным видео в сравнении с другими моделями в сценарии черного ящика. В обоих сценариях атака на модели автокодировщика затрагивает большие области изображения даже при небольшом размере патча (0,1 % от разрешения изображения). В то же время некоторые другие типы моделей более устойчивы к такой атаке.

15.7. КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ

В этом разделе мы обсудим и систематизируем обманные образцы для задачи классификации изображений.

15.7.1. Белый ящик, ограниченные атаки

Ограниченные атаки на белый ящик обычно генерируют возмущения с ограничением ℓ_p -нормы, так что искажение полученного состязательного образца незаметно для людей.

L-BFGS (Szegedy et al., 2014), чтобы ввести в заблуждение классификатор, использует обманные изображения, которые состоят из незаметных состязательных возмущений, добавленных к нормализованному входу $\mathbf{x} \in [0, 1]$. Атака L-BFGS решает задачу оптимизации с ограничениями, целью которой является нахождение $\dot{\mathbf{x}} \in [0, 1]$ с ограничением по норме ℓ_2 :

$$c\|\dot{\mathbf{x}} - \mathbf{x}\|_2 + \mathcal{L}(\dot{\mathbf{x}}, \dot{y}), \quad (15.6)$$

где c – это гиперпараметр, а $\mathcal{L}(\dot{\mathbf{x}}, \dot{y})$ – потеря перекрестной энтропии, которая измеряет разницу между меткой y , предсказанной целевым классификатором на основе обманных входных данных, и меткой целевой ошибочной классификации \dot{y} . Эта потеря побуждает атаку генерировать $\dot{\mathbf{x}}$, способный ввести в заблуждение целевой классификатор. Атака Карлини–Вагнера (CW) (Carlini, Wagner, 2017) генерирует возмущение, ограниченное ℓ_p -нормой, путем решения задачи оптимизации с ограничениями, аналогичной L-BFGS. Эта атака минимизирует потери в рамках ограничений ℓ_p -нормы, которые измеряют разницу между логит-значением \dot{z}_y образца $\dot{\mathbf{x}}$, принадлежащим тому же классу y предсказания по \mathbf{x} , и максимальным логит-значением среди всех остальных классов:

$$\min_{\delta} \left(\|\dot{\mathbf{x}} - \mathbf{x}\|_p + c \left(\max_{n=1, \dots, D} \{\dot{z}_n; n \neq y\} - \dot{z}_y \right) \right), \quad (15.7)$$

где D – общее количество меток $p \in \{0, 2, \infty\}$, а $c > 0$ – константа, определяемая с помощью линейного поиска для нахождения оптимального значения. Минимизация второго члена побуждает атаку находить образец $\dot{\mathbf{x}}$, который делает $\dot{z}_n \geq \dot{z}_y$, чтобы в результате классификации можно было избежать y . В отличие от L-BFGS, в котором используется кросс-энтропийная потеря, CW-атака основана на маржинальной потере, которая более эффективна при поиске минимально искаженного обманного образца (Carlini, Wagner, 2017).

Знаковый метод быстрого градиента (fast gradient sign method, FGSM) (Goodfellow et al., 2015) основан на градиенте и определяет направление возмущения таким образом, что потери в целевой модели увеличиваются. FGSM оценивает возмущения, вычисляя градиент функции потерь, $\mathcal{L}(\mathbf{x}, y)$ по отношению к заданному входу \mathbf{x} , с небольшим ϵ для создания незаметных вредоносных возмущений:

$$\dot{\mathbf{x}} = \mathbf{x} + \epsilon \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, y)). \quad (15.8)$$

Поскольку возмущения генерируются в направлении $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, y)$, обманное изображение может ввести целевую модель в заблуждение таким образом, чтобы она избежала исходную метку y , что приведет к ненаправленной атаке. FGSM также можно использовать для направленной атаки, уменьшив потери целевой модели по отношению к целевой метке \dot{y} как

$$\dot{\mathbf{x}} = \mathbf{x} - \epsilon \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \dot{y})), \quad (15.9)$$

что создает возмущение, которое вводит классификатор в заблуждение и заставляет его выбрать целевую метку \dot{y} .

Варианты FGSM, использующие оптимизацию на основе градиента, также могут быть либо ненаправленными, либо направленными аналогичным образом, как показано в уравнениях (15.8) и (15.9). **Базовый итерационный метод** (BIM или I-FGSM) (Kurakin et al., 2017) расширяет FGSM путем агрегирования обманных возмущений для фиксированного числа итераций:

$$\dot{\mathbf{x}}_{i+1} = \mathcal{C}_{\dot{\mathbf{x}}_i, \epsilon}(\dot{\mathbf{x}}_i + \alpha \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(\dot{\mathbf{x}}_i, y))), \quad (15.10)$$

где α управляет величиной возмущений на каждом шаге, а операция отсечения $\mathcal{C}_{\dot{\mathbf{x}}_i, \epsilon}(\cdot)$ обрезает интенсивности пикселей обманного изображения на временном шаге i , чтобы они находились в диапазоне исходного изображения. **Проецируемый градиентный спуск** (PGD) (Madry et al., 2018) обобщает BIM без ограничений на величину возмущения. Вместо этого на каждом шаге PGD проецирует обманные образцы на ℓ_{∞} -соседа, чтобы можно было ограничить возмущение:

$$\dot{\mathbf{x}}_{i+1} = \mathcal{P}_{\dot{\mathbf{x}}_i, \epsilon}(\dot{\mathbf{x}}_i + \alpha \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(\dot{\mathbf{x}}_i, y))), \quad (15.11)$$

где $\mathcal{P}_{\dot{\mathbf{x}}_i, \epsilon}$ – оператор проецирования. **Атаки на основе якобиановой карты значимости** (Jacobian-based saliency map attacks, JSMA) (Papernot et al., 2016) создают карту значимости $S(\mathbf{x}_{\mathbf{q}}, \dot{y})$, которая определяет каждый пиксель $x_{\mathbf{q}} \in \mathbf{x}$ в позиции \mathbf{q} , чтобы выбрать пиксели, которые наиболее вероятно придется нарушить для достижения желаемых изменений в классификации целевой метки \dot{y} :

$$S(x_q, \dot{y}) = \begin{cases} 0, & \nabla_{x_q} P_{\dot{y}}(\mathbf{x}) < 0 \text{ или } \sum_{j \neq \dot{y}} \nabla_{x_q} P_j(\mathbf{x}) > 0 \\ \nabla_{x_q} P_{\dot{y}}(\mathbf{x}) \left\| \sum_{j \neq \dot{y}} \nabla_{x_q} P_j(\mathbf{x}) \right\|_1 & \text{в ином случае} \end{cases}, \quad (15.12)$$

где $P_{\dot{y}}(\mathbf{x})$ – предсказанная вероятность softmax для \dot{y} до последнего слоя классификации. Алгоритм возмущает пиксель x_q с наибольшим значением $S(x_q, \dot{y})$, чтобы увеличить (или уменьшить) вероятности softmax целевого класса. Атака итеративно генерирует возмущение до тех пор, пока злонамеренное изображение не будет классифицировано как \dot{y} или пока не будет нарушено заданное максимальное количество пикселей.

DeepFool (Moosavi-Dezfooli et al., 2016) – это еще один подход, основанный на градиенте, который генерирует возмущения $\dot{\mathbf{x}}_i$, ограниченные l_2 , путем оценки расстояния от входа на временном шаге i до ближайшей границы решения целевого классификатора с оригинальным классом y . Этот процесс повторяется до тех пор, пока $f(\dot{\mathbf{x}}_i) \neq f(\mathbf{x}_i)$. DeepFool производит меньшие возмущения по сравнению с атакой L-BFGS и с меньшими затратами на вычисления. **SparseFool** (Modas et al., 2019) использует состязательную атаку, подобную DeepFool, с ограничением возмущений l_1 -нормой. SparseFool использует малый средний радиус кривизны границы решения для эффективного вычисления обманных возмущений с помощью нескольких пикселей.

Вышеупомянутые методы обычно сосредоточены на *эффективности* и *заметности* атаки со стороны противника, в то время как *переносимости* уделяется меньше внимания. Некоторые варианты FGSM также допускают дополнительные проблемы со стороны злоумышленников, такие как введение в заблуждение нескольких моделей или незнакомых моделей и защиту частной информации, например исходного класса, из входного изображения. Возмущения, созданные в результате атак на основе FGSM, могут быть адаптированы к целевой модели. Чтобы улучшить возможность переноса на несколько моделей, вариант Diverse Input Iterative FGSM (DI2-FGSM) (Xie et al., 2019) применяет случайное изменение размера и дополнение входных изображений на каждой итерации для создания жестких и разнообразных входных шаблонов. Ансамбль FGSM (E-FGSM) (Tramèr et al., 2018) использует несколько классификаторов одновременно при создании возмущений, подобных FGSM (уравнение 15.9). E-FGSM может быть применим к нескольким моделям, которые используются для обучения, но возмущения, как правило, переопределяются для используемых моделей и плохо подходят для незнакомых классификаторов. Заметим, что из этих вариантов FGSM можно легко вывести истинный класс изображений, поэтому важно защитить конфиденциальность входной метки. **Частный FGSM** (P-FGSM) (Li et al., 2019) достигает этого, отбрасывая лучшие предсказанные классы из выбора класса, чтобы создать возмущение, подобное уравнению (15.9), но не рассматривает возможность переноса на другие модели или средства защиты, которые могут оценить исходную метку. Чтобы удовлетворить как проблемы переносимости, так и конфиденциальности, а также обнаруживаемости при состязательной атаке, **Robust Private FGSM** (RP-FGSM) (Sanchez-Matilla et al., 2020) случайным образом выбирает целевую модель для атаки, а также защиту, от которой нужно уклониться, на основе стратегии P-FGSM для выбора класса.

В отличие от предыдущих атак, которые создают возмущения на конкретном изображении (атаки, зависящие от изображения), **универсальное состязательное возмущение** (universal adversarial perturbation, UAP) (Moosavi-Dezfooli et al., 2017) представляет собой единственное возмущение, целью которого является введение модели в заблуждение для большинства изображений в обучающем наборе. UAP накапливает возмущения, зависящие от изображения, итеративно применяя DeepFool до тех пор, пока определенная часть входных изображений не будет классифицирована неправильно. Таким образом, UAP требует обучающих данных, на которых целевая модель может быть обучена создавать единственное универсальное возмущение. **Fast Feature Fool** (Mopuri et al., 2017) предполагает отсутствие доступа к исходным обучающим данным. Этот метод направлен на создание универсальных возмущений, которые могут обмануть признаки целевой модели, оптимизируя произведение средних активаций на нескольких уровнях целевой модели, когда входными данными являются универсальные возмущения.

Обсуждаемые до сих пор состязательные атаки генерируют возмущения, решая задачу оптимизации с ограничениями или обновляя обратную связь от градиентов по отношению к входным изображениям. В отличие от вышеупомянутых методов, **сети состязательного преобразования** (adversarial transformation networks ATN) (Baluja, Fischer, 2018) обучают нейронную сеть с прямой связью создавать обманные изображения, похожие на входное изображение. После обучения сеть может генерировать их быстрее, чем алгоритмы на основе оптимизации и алгоритмы на основе градиента с итеративными обновлениями. Точно так же **обманные генеративно-состязательные сети** (AdvGAN) (Xiao et al., 2018) учатся создавать злонамеренные возмущения с использованием нейронных сетей, созданных по принципу генеративно-состязательной архитектуры (Goodfellow et al., 2014), которые состоят из генератора и дискриминатора. Генератор в AdvGAN генерирует возмущения, тогда как дискриминатор, обученный отличать реальное входное изображение от обманных изображений, учит генератор создавать обманные изображения, максимально похожие на входное изображение. Этот метод можно рассматривать как атаку «серого ящика», поскольку обученная сеть прямого распространения (генератор) может создавать обманные возмущения для любых входных данных, не требуя больше доступа к самой модели. В этой архитектуре атака черного ящика также доступна путем замены целевой модели дистиллированной (или замещающей) моделью. **Генеративно-состязательные возмущения** (generative adversarial perturbations, GAP) (Poursaied et al., 2018) – это генеративная модель, которая создает зависящие от изображения или универсальные возмущения для ненаправленных или направленных атак. Возмущения, зависящие от изображения, создаются аналогично AdvGAN (Xiao et al., 2018), где генератор формирует возмущение, исходя из входного изображения. Для выработки универсального возмущения генератор берет фиксированный шаблон, выбранный из равномерного распределения, чтобы создать возмущение, которое добавляется к чистым изображениям и обманывает классификатор. **Сеть злонамеренной генерации** (network for adversary generation, NAG) (Mopuri et al., 2018) создает универсальные возмущения, используя GAN-подобную генеративную модель,

которая моделирует неизвестное распределение обманных возмущений для данного классификатора DNN. Основываясь на предполагаемом распределении, NAG может генерировать широкий спектр универсальных возмущений.

15.7.2. Белый ящик, атаки на основе контента

Возмущения на основе контента создаются с учетом свойств изображения, таких как структура изображения (Shamsabadi et al., 2020a), текстуры или цвета (Bhattad et al., 2020). В отличие от возмущений, ограниченных нормой (Goodfellow et al., 2014; Szegedy et al., 2014), возмущения, основанные на контенте, являются неограниченными, т. е. не ограничивают величину обманных возмущений. Это позволяет состязательным атакам на основе контента улучшить переносимость и снизить обнаруживаемость для средств защиты.

Семантическая манипуляция (semantic manipulation) (Bhattad et al., 2020) представляет собой две состязательные атаки, которые манипулируют визуальными дескрипторами, а именно цветами и текстурами входного изображения. Цветовая стратегия использует в качестве направления состязательной атаки раскрашивание изображения. Имея входное изображение, преобразованное в оттенки серого, злоумышленник учится раскрашивать его с помощью исходных цветов исходного изображения таким образом, чтобы ввести в заблуждение целевой классификатор. Текстурная атака переносит текстуру с другого изображения на входное изображение. **EdgeFool** (Shamsabadi et al., 2020) создает обманные изображения с улучшенной детализацией. EdgeFool обучает нейронную сеть с прямой связью с многозадачными функциями потерь, которые совместно учитывают сглаживание изображения и ненаправленную состязательную атаку (Carlini, Wagner, 2017). Детали изображения, извлеченные из изученного гладкого изображения, обрабатываются с помощью традиционной техники повышения детализации (Farbman et al., 2008). Выходные данные улучшения детализации затем передаются в функцию потерь CW (Carlini, Wagner, 2017), чтобы платформа EdgeFool могла генерировать обманные изображения с улучшенной детализацией.

15.7.3. Атаки методом черного ящика

Атаки методом черного ящика предполагают наличие ограниченной информации или отсутствие информации о целевой модели, например только окончательную выходную метку или оценки прогноза. Атаку черного ящика можно выполнить путем обучения модели-заменителя, которая способна аппроксимировать целевую модель (аппроксимация границы решения), генерируя возмущение на основе предполагаемого градиента потерь с помощью запросов, которые возвращают оценки или вероятности меток из целевой модели (оценка градиента), манипулируя возмущениями текущего шага в правильном направлении, чтобы ввести в заблуждение целевую модель (локальный поиск), или выполняя жадный поиск до тех пор, пока целевая модель не будет введена в заблуждение или не достигнет максимальной итерации (случайный поиск).

Методы *аппроксимации границы решения* обычно учитывают переносимость обманных изображений. Атака замещающего черного ящика (substitute blackbox attack, SBA) (Papernot et al., 2016) обучает замещающую модель, которая имитирует исходную модель, а затем использует атаки белого ящика на обученной замещающей модели для создания обманных возмущений. Используя возможность переноса обманных изображений, SBA получает прогнозы для синтетического набора данных из целевой модели, а затем обучает замещающую модель, чтобы имитировать прогноз целевой модели. Обученная замещающая модель используется в качестве псевдоцелевой модели, где злоумышленники теперь могут генерировать возмущения с помощью атаки белого ящика. Поскольку замещающая модель обучается на основе предположения о переносимости возмущений, важно выбирать атаки, обладающие высокой переносимостью. Атака **Curls & Whey** (Shi et al., 2019) направлена на улучшение переносимости обманных изображений на другие целевые модели. Во-первых, итерация Curls создает возмущения вдоль направления градиентного подъема и спуска замещающей модели, что позволяет обманным изображениям иметь лучшую переносимость за счет разнообразия сгенерированных изображений. Второй шаг, оптимизация Whey, уменьшает величину возмущений от созданных обманных изображений. Поскольку известно, что хорошая переносимость обманных изображений позволяет проводить атаки черного ящика, в атаке **Translation-Invariant** (TI) (Dong et al., 2019) используется ансамбль перенесенных обманных изображений, а не оптимизация функции потерь для оценки заменяющей модели. Метод TI выполняет атаку черного ящика путем создания переносимых обманных изображений, которые генерируются для другого классификатора белого ящика со свойством инвариантности к переносу, но имеют высокую переносимость для атак черного ящика.

Атаки путем оценки градиента принимают форму запроса обратной связи, в которой злоумышленник итеративно генерирует возмущения и спрашивает, не ошиблась ли модель-жертва, до тех пор, пока цель не будет достигнута. Оценивая градиент функции потерь из входных запросов, злоумышленник может генерировать возмущение на основе направления этого градиента. **Оптимизация нулевого порядка** (zeroth order optimization, ZOO) (Chen et al., 2017) оценивает градиенты целевой модели на основе предположения, что злоумышленник имеет доступ к получению оценок вероятности всех классов из целевой модели. Это предположение легче реализует атаку черного ящика, чем предыдущие работы (Papernot et al., 2016), которые могут получить только информацию о метке из классификатора. ZOO генерирует состязательные возмущения, наблюдая за изменениями вероятностей, что позволяет аппроксимировать градиенты. Затем аппроксимированные градиенты можно использовать для стохастического спуска по координатам при выполнении состязательной атаки. Одной из проблем атаки методом «черного ящика» является большое количество запросов, которые нужны для прогнозирования неизвестных целевых моделей. **Атака с ограничением запросов** (Ilyas et al., 2018) генерирует обманные образцы с ограниченным количеством запросов. Метод использует стратегии естественной эволюции (Salimans et al., 2017) в качестве способа оценки градиента при атаке черного

ящика. Имея предполагаемый градиент, можно вместе с PGD (Madry et al., 2018), используемым для атак белого ящика, создать обманное изображение. **Оптимизация нулевого порядка на основе автоэнкодера** (autoencoder based zeroth order optimization, AutoZOOM) (Tu et al., 2019) также решает проблемы эффективности запросов. AutoZoom уменьшает количество запросов, необходимых для создания успешных обманных изображений, с помощью стратегии адаптивной оценки случайного градиента, в которой атака может быть ускорена с помощью автокодировщика, обученного в автономном режиме с немаркированными данными, или билинейной операции изменения размера.

Методы случайного поиска создают случайные возмущения до тех пор, пока полученное изображение не введет целевую модель в заблуждение. **SemanticAdv** (Hosseini, Poovendran, 2018) изменяет входное изображение в цветовом пространстве HSV. Оттенок и насыщенность входного изображения искажаются случайным возмущением, равномерно выбираемым из диапазона допустимых значений. SemanticAdv вносит новые возмущения до тех пор, пока классификатор не будет введен в заблуждение или не будет достигнуто максимальное количество попыток (например, 1000). Поскольку значения оттенка и насыщенности меняются на одинаковую величину, обманные изображения могут выглядеть неестественно. В отличие от SemanticAdv, модель **ColorFool** (Shamsabadi et al., 2020) генерирует возмущения, которые, кроме быстрого действия, учитывают естественность получаемых обманных изображений. Метод сначала находит нечувствительные и чувствительные области посредством семантической сегментации. Затем возмущения воздействуют на каналы *a* и *b* цветового пространства *Lab*: возмущения нечувствительных областей выбираются случайным образом из всего диапазона возможных значений, тогда как возмущения чувствительных областей выбираются случайным образом из predetermined диапазонов естественных цветов, ориентированных на человеческое восприятие. Кроме того, SemanticAdv и ColorFool создают неограниченные возмущения, которые обеспечивают лучшую переносимость на другие модели, чем другие атаки черного ящика, которые в основном ограничивают возмущения, уделяя особое внимание разработке методологии атаки.

15.8. СЕМАНТИЧЕСКАЯ СЕГМЕНТАЦИЯ И ОБНАРУЖЕНИЕ ОБЪЕКТОВ

В этом разделе мы обсудим состязательные атаки, разработанные для моделей семантической сегментации и обнаружения объектов. Семантическая сегментация выполняет маркировку сегментов объекта, присваивая метку каждому пикселю изображения. Обнаружение объектов, в отличие от семантической сегментации, локализует объекты с помощью ограничивающих рамок и классифицирует каждый объект.

Атака семантической сегментации (semantic segmentation attack, SSA) (Fischer et al., 2017) использует атаку BIM (Kurakin et al., 2017) для каждого

пикселя изображения или области с определенной меткой, чтобы ввести в заблуждение задачу попиксельной семантической сегментации с помощью FCN. **Плотная состязательная генерация** (dense adversary generation, DAG) (Xie et al., 2017) создает состязательные возмущения для задач обнаружения объектов и семантической сегментации, выполняемых сетями Faster R-CNN и FCN соответственно. Обратите внимание, что в одном изображении Faster R-CNN выполняет классификацию нескольких объектов-кандидатов, чтобы определить местонахождение объекта в виде ограничивающей рамки, а FCN классифицирует метку каждого пикселя в изображении. Рассматривая объекты-кандидаты и пиксели изображения как несколько целей, которые можно ввести в заблуждение, DAG создает возмущения на основе градиента, стремящиеся ввести в заблуждение как можно больше пикселей. **Универсальная эффективная состязательная сеть** (Unified and Efficiency Adversary, UEA) (Wei et al., 2019) ненаправленно генерирует обманные изображения с помощью условной генеративно-состязательной сети (сGAN), чтобы вводить в заблуждение детекторы объектов. UEA вместе с состязательной потерей, используемой для атаки на сети Faster R-CNN в DAG, выводит функцию потерь, которая интегрирует многомасштабную потерю функции внимания, чтобы лучше сконцентрировать возмущения на областях объекта. Изображения, созданные с многомасштабным вниманием, могут ввести в заблуждение Faster R-CNN с меньшим количеством возмущений, чем DAG, и улучшить переносимость атаки черного ящика на другой детектор объектов, например SSD. Детектор границы объекта может быть атакован, как показано в **Edge Attack** (Cosgrove, Yuille, 2020), в которой используются варианты FGSM (Goodfellow et al., 2015), чтобы ввести в заблуждение детектор HED (Xie, Tu, 2015) (раздел 15.6). Edge Attack передает созданные обманные изображения на последующую классификацию изображений и семантическую сегментацию.

15.9. ОТСЛЕЖИВАНИЕ ОБЪЕКТА

В моделях DNN отслеживание отдельных объектов рассматривается как задача метрики подобию на основе шаблона, которая ищет в каждом кадре область, наиболее похожую на эталонный патч, заданный как приор или выбранный из первого кадра. В этой задаче используются сиамские архитектуры (Bromley et al., 1994) для сравнения эталонного участка с участком из области поиска в кадре (Bertinetto et al., 2016; Li et al., 2018; Zhang, Peng, 2019; Li et al., 2019). Состязательная атака на систему отслеживания объекта может генерировать возмущения, которые изменяют эталонный патч, область поиска или и то, и другое.

Быстрая атакующая сеть (fast attack network, FAN) (Liang et al., 2020) на сегодняшний день является единственным инструментом состязательной атаки на систему отслеживания объектов. FAN может выполнять ненаправленную или направленную атаку. Ненаправленная FAN возмущает область поиска независимо для каждого кадра последовательности, так что карта отклика максимизируется в случайных областях вне истинной траектории

объекта. Направленная FAN побуждает трекер следовать по другой, заранее определенной траектории, нарушая как эталонный участок, так и области поиска. Чтобы избежать преждевременных отказов трекера при целевой атаке, FAN модифицирует области поиска (опорный патч) таким образом, чтобы расстояние в пространстве признаков между обманной областью поиска (или обманным эталонным патчем) и заданными областями траектории было минимальным.

FAN использует генеративный подход для получения возмущений, незаметных для человеческого глаза, а также легко добавляемых к входным видео. Во время обучения параметры FAN оптимизируются путем чередования генератора и дискриминатора. Функция потерь для дискриминатора $\mathcal{D}(\cdot)$ основана на PatchGAN (Isola et al., 2017). Генератор $\mathcal{G}(\cdot)$ обучается с помощью многокритериальной функции потерь \mathcal{L}_F , которая учитывает расстояние между представлениями признаков \mathcal{L}_e (целевая атака); двучелевой член дрейфа $\beta_1 \mathcal{L}_d + \beta_2 \mathcal{L}_s$ (ненаправленная атака); расстояние по норме ℓ_2 между исходным и обманным изображениями \mathcal{L}_u (незаметность) в дополнение к стандартному члену генератора \mathcal{L}_G (на основе циклической GAN):

$$\mathcal{L}_F = \mathcal{L}_G + \alpha_1 \mathcal{L}_u + \alpha_2 \mathcal{L}_e + \alpha_3 (\beta_1 \mathcal{L}_d + \beta_2 \mathcal{L}_s), \quad (15.13)$$

где α_1 , α_2 , α_3 , β_1 и β_2 – гиперпараметры для балансировки влияния каждого члена многокритериальной потери. Гиперпараметры для направленной и ненаправленной атак обнуляются при оптимизации любой из атак. Член потерь для генератора определяется как

$$\mathcal{L}_G = \mathbb{E}[\mathcal{D}(\delta + R(\mathbf{x}_k, \mathbf{b}_k)) - 1]^2, \quad (15.14)$$

где \mathbf{x}_t – исходный видеокادر в момент времени k , $R(\cdot)$ – экстрактор региона на основе области поиска \mathbf{b}_k в кадре k . Чтобы создать незаметное возмущение, член подобия в уравнении потерь определяется как

$$\mathcal{L}_u = \mathbb{E}_{\mathbf{x}_k \sim \mathcal{X}} [\|R(\dot{\mathbf{x}}_k, \mathbf{b}_k) - R(\mathbf{x}_k, \mathbf{b}_k)\|_2], \quad (15.15)$$

где \mathcal{X} обозначает набор кадров во входном видео. При направленной атаке \mathcal{L}_e стремится минимизировать расстояние в пространстве признаков между обманным опорным патчем и заданной траекторией

$$\mathcal{L}_e = \mathbb{E}_{\mathbf{x}_k \sim \mathcal{X}, \dot{\mathbf{x}}_k \sim \mathcal{E}} [\|\phi(R(\mathbf{x}_k, \mathbf{b}_k) + \delta) - \phi(\dot{\mathbf{x}}_k)\|_2], \quad (15.16)$$

где $\dot{\mathbf{x}}_k$ – область заданной обманной траектории \mathcal{E} в кадре k , $\phi(\cdot)$ – функция, которая отображает область изображения во встроенное пространство признаков, а δ генерируется $\mathcal{G}(\cdot)$ и ограничивается до области изображения, выбранной с помощью $R(\cdot)$. Обратите внимание, что $R(\mathbf{x}_k, \mathbf{b}_k)$ является опорным патчем, когда $k = 0$.

При ненаправленной атаке член дрейфа нацелен на максимизацию оценки отклика за пределами области, где трекер обычно имеет самый высокий отклик в пределах области поиска для отслеживания объекта:

$$\mathcal{L}_d = \frac{1}{\gamma + \|\mathbf{q}_{\max}^{+1} - \mathbf{q}_{\max}^{-1}\|_2} - \xi, \quad (15.17)$$

в то же время максимально отдаляя фиктивный центр активации от реального:

$$\mathcal{L}_s = \min_{\mathbf{q} \in \mathcal{S}^{+1}} (\log(1 + e^{y_q s_q})) - \min_{\mathbf{q} \in \mathcal{S}^{-1}} (\log(1 + e^{y_q s_q})), \quad (15.18)$$

где $\mathcal{S} = \mathcal{S}^{+1} \cup \mathcal{S}^{-1}$ – карта отклика с положительными и отрицательными метками $y_q = \{-1, +1\}$ для каждого положения пикселя \mathbf{q} ; \mathbf{q}_{\max}^{+1} и \mathbf{q}_{\max}^{-1} – расположение пикселей с максимальными показателями активации в положительных и отрицательных областях карты откликов соответственно; s_q – оценка отклика в точке расположения пикселя \mathbf{q} ; γ – малая константа для числовой стойкости, а ξ управляет степенью смещения центра активации.

15.10. Классификация видео

Атаки на классификатор изображений можно естественным образом распространить на временную область, чтобы ввести в заблуждение целевую модель для классификации видео, работая с каждым кадром независимо. Тем не менее временную информацию можно использовать для создания более надежных злонамеренных возмущений, используя пакет кадров или обрабатывая кадры в режиме реального времени по мере их получения камерой. В этом разделе мы обсудим три составительные атаки на системы классификации видео: C-DUP, V-BAD и MultAV.

Круговые универсальные возмущения двойного назначения (circular universal dual purpose perturbations, C-DUP) – это атака белого ящика, которая вводит в заблуждение классификаторы видео, работающие в реальном времени (Li et al., 2019b). C-DUP направлена на то, чтобы вызвать неправильную классификацию только определенных классов, сохраняя при этом неизменным распознавание других классов. C-DUP расширяет работу Морпури для статичных изображений (Morpurì et al., 2018) и использует модифицированную GAN для изучения в автономном режиме единого набора (универсальных) возмущений, которые можно применить в режиме реального времени к незнакомым входным данным. GAN исправляет дискриминатор с помощью известного предварительно обученного классификатора и лишь обучает 3D-генератор создавать универсальные возмущения, которые добавляются к входным видеокадрам, стремясь обмануть дискриминатор. Чтобы атаковать только определенные классы, C-DUP использует функцию потерь с векторами вероятности всех классов, выдаваемыми дискриминатором, по всем обучающим данным. Эта функция потерь двойного назначения поддерживает минимизацию кросс-энтропии для целевого класса(ов) $\hat{y} \in \mathcal{A} \subset \mathcal{Y}$ (где \mathcal{Y} – набор меток классов) и максимизацию кросс-энтропии для всех других классов $\mathcal{Y} \setminus \mathcal{A}$:

$$\min_{\mathcal{G}} \sum_{o=1}^W \left(\sum_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_a} -\log[P_y(\mathbf{x} + \rho(\mathcal{G}(\mathbf{h}), o))] \right. \\ \left. + \lambda \sum_{\mathbf{x}_a \in \mathcal{X}_a} -\log[1 - P_y(\mathbf{x}_a + \rho(\mathcal{G}(\mathbf{h}), o))] \right), \quad (15.19)$$

где λ – весовой параметр, уравнивающий потери, \mathbf{h} – вектор шума из скрытого пространства (например, равномерное распределение в диапазоне $[-1, 1]$); $G(\cdot)$ – генератор; P_y – вероятность неатакованной метки y ; $P_{\tilde{y}}$ – вероятность атакованной метки \tilde{y} ; $\mathcal{X}_a \subset \mathcal{X}$ – подмножество видеоклипов, метка действия которых подвергается атаке; и $\rho(G(\mathbf{h}), o) = \delta_o$ – функция перестановки, которая применяет циклический сдвиг по всем кадрам ($o = 1, \dots, W$, где W – количество кадров в клипе) сгенерированного возмущения. Функция перестановки заставляет C-DUP быть независимой от временной последовательности кадров в клипе. Для создания этих круговых возмущений между 3D-генератором и дискриминатором используется блок постобработки. Наконец, 3D-генератор ограничивает возмущения, чтобы они находились в пределах единичного шара, определяемого границей ϵ . Успех C-DUP на грубых (UCF) и детализованных (JESTER) наборах данных распознавания действий для введения в заблуждение пространственно-временного классификатора C3D (Tran et al., 2015) превышает 80 % для целевых классов при сохранении точности классификации выше 80 % для других классов (исходная точность классификации для C3D составляет 96 % и 90 % для UCF и JESTER соответственно). Существует вариант C-DUP, который генерирует одно квазинезаметное двумерное возмущение, которое применяется к каждому кадру клипа. Однако эта атака менее эффективна, чем C-DUP, особенно для детализованной классификации действий, где более важна информация об изменениях по оси времени.

V-BAD – это атака методом черного ящика с трехэтапной итеративной схемой для запроса атакуемой модели и получения для каждой итерации метки классификации и соответствующей вероятности (Jiang et al., 2019). Атака V-BAD может быть направленной или ненаправленной. Направленная атака V-BAD нацелена на то, чтобы обмануть модель с помощью целевого класса, возвращаемого как top-1, и использует образец видео из целевого класса в качестве входных данных, регулируя размер границы возмущения. Ненаправленная атака V-BAD использует исходное видео в качестве входных данных и сохраняет размер возмущения постоянным. Возмущения незаметны и ограничены ϵ -шаром с центром в исходном видео. Предварительные возмущения генерируются независимо для каждого кадра исходного входного видео (или обманного образца для каждой итерации) с помощью ансамбля из трех предварительно обученных DNN, а затем их выходные данные усредняются. V-BAD использует равномерные разделы в патчах входных возмущений и ограничивает количество запросов к атакуемой модели распознавания видео с помощью оценки градиента по патчам. Оценщик градиента предоставляет вектор весов, основанный на состязательных потерях относительно вероятности, возвращенной атакуемой моделью. Затем патчи возмущений исправляются с использованием вектора весов и применяются к входному видео (или обманному изображению из предыдущей итерации) с помощью одношагового PGD. V-BAD использовался для атаки на классификаторы видео CNN+LSTM и I3D.

MultAV представляет собой набор прямых или итерационных атак на основе градиента (ℓ_p -норма, ℓ_2 -норма PGD), которые создают для каждо-

го кадра видео *мультипликативные* возмущения (Lo, Patel, 2020). Подобно их аналогу с аддитивным возмущением, атаки MultAV на основе градиента ограничены. В частности, MultAV вводит граничную оценку ϵ_m (где m означает мультипликативное), которая ограничивает попиксельное соотношение между обманным и исходным изображениями, чтобы сделать искажение незаметным. Если рассматривать в качестве примера ограниченную атаку по норме ℓ_2 с аддитивным возмущением (уравнение 15.10), то ее мультипликативный аналог, создаваемый итеративным MultAV- ℓ_2 , генерируется следующим образом. Обманное изображение на итерации $i + 1$ получается путем перемножения обманного изображения из предыдущей итерации с размером мультипликативного шага α_m , показатель степени которого определяется градиентом потерь

$$\mathbf{x}_{i+1} = \mathcal{C}_{\mathbf{x}_i, \epsilon_m} \left\{ \mathbf{x} \odot \alpha_m^{\frac{\nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i, \mathbf{y})}{\|\nabla_{\mathbf{x}_i} \mathcal{L}(\mathbf{x}_i, \mathbf{y})\|_2}} \right\}, \quad (15.20)$$

где $\mathcal{C}_{\mathbf{x}_i, \epsilon_m} \{\cdot\}$ – операция отсечения, ϵ_m – ограниченное отношение, такое что $\left\| \frac{\mathbf{x}_i + 1}{\mathbf{x}} \right\|_2 < (\epsilon_m + 1)$, а \odot – произведение Адамара (поэлементное). В качестве альтернативы MultAV применяет тот же мультипликативный принцип к методам на основе патчей, таким как прямоугольная окклюзия и состязательное кадрирование, или к аддитивному шуму на основе пикселей. Однако эти атаки приводят к ощутимым возмущениям. MultAV был применен на UCF, чтобы ввести в заблуждение классификатор видео 3D ResNet-18.

15.11. ЗАЩИТА ОТ СОСТЯЗАТЕЛЬНЫХ АТАК ПРОТИВНИКА

Существуют специальные способы защиты моделей машинного обучения от атак злоумышленников. Защита может обнаружить обманное изображение (защита путем обнаружения) или исказить градиентную обратную связь от функции потерь в сети, чтобы отключить атаку (защита с помощью градиентной маскировки). В качестве альтернативы целевую модель можно обучить на обманных образцах, чтобы повысить ее устойчивость к атакам (надежность модели). Способы защиты и их свойства перечислены в табл. 15.3 и обсуждаются ниже.

15.11.1. Обнаружение атаки

Различные способы защиты путем обнаружения атаки направлены на то, чтобы отличить обманное изображение и отвергнуть его. Эти средства защиты могут использовать статистику, полагаться на создание вспомогательной модели или на анализ *согласованности* прогнозов.

Таблица 15.3. Сводная информация о средствах защиты от состязательных атак. Обозначения: Dec – метод на основе обнаружения, GradM – маскирование градиента, ModelR – надежность модели, Aux – вспомогательная модель, Stat – статистический подход, CCheck – проверка целостности, NonDiff – недифференцируемый градиент, Van/Exp – исчезающий/взрывающийся градиент, Stoch – стохастический градиент, AdvT Adversarial training, Reg – регуляризация, Cert – сертифицированная защита, Exp – эксперимент, C – классификация изображений, D – обнаружение объектов, S – семантическая сегментация, F – распознавание лиц, T – отслеживание объектов

Источник	Метод	Цель	Подход	Наборы данных	Задача
Hendrycks and Gimpel (2016)	PCA	Dec	Stat	ImageNet, MNIST, CIFAR	C
Grosse et al. (2017)	Статистический тест	Dec	Stat	MNIST	C
Gong et al. (2017)	Бинарный классификатор	Dec	Aux	MNIST, CIFAR, SVHN	C
Metzen et al. (2017)	Сеть обнаружения обмана	Dec	Aux	ImageNet, CIFAR	C
Feinman et al. (2017)	KDE & BUE	Dec	CCheck	MNIST, CIFAR	C
Xu et al. (2018)	Уплотнение признаков	Dec	CCheck	MNIST, CIFAR	C
Buckman et al. (2018)	Термометрическое кодирование	GradM	NonDiff	MNIST, CIFAR, SVHN	C
Guo et al. (2018)	Преобразования изображения	GradM	NonDiff	ImageNet	C
Papemot et al. (2016с)	Защитная дистилляция	GradM	Van/Exp	MNIST, CIFAR	C
Song et al. (2018)	Пиксельная защита	GradM	Van/Exp	F-MNIST, CIFAR	C
Samangouei et al. (2018)	Защитная GAN	GradM	Van/Exp	F-MNIST	C
Zhou et al. (2020)	A-VAE	GradM	Van/Exp	LFW	F
Dhillon et al. (2018)	Стохастическое отсечение	GradM	Stoch	CIFAR	C
Xie et al. (2018)	Рандомизация	GradM	Stoch	ImageNet	C
Gu, Rigazio (2014)	Глубокая сокращаемая сеть	ModelR	Reg	MNIST	C
Cisse et al. (2017)	Сеть Парсеваля	ModelR	Reg	CIFAR, SVHN	C
Goodfellow et al. (2015)	Базовая AdvTrain	ModelR	AdvT	MNIST	C
Madry et al. (2018)	PGD AdvTrain	ModelR	AdvT	MNIST, CIFAR	C
Tramer et al. (2018)	Ансамблевая AdvTrain	ModelR	AdvT	ImageNet	C
Zhang, Wang (2019)	AROD	ModelR	AdvT	PascalVOC, COCO	D
Jia et al. (2020)	RT	ModelR	AdvT	OTB, VOT, UAV	T
Raghuathan et al. (2018)	Одиночное квазиопределение	ModelR	Cert	MNIST	C
Wong, Kolter (2018)	Глубокая ReLU	ModelR	Cert	F-MNIST, HAR, SVHN	C
Amab et al. (2018)	RSSM	ModelR	Exp	PascalVOC, CityScapes	S

Обманные изображения можно обнаружить, найдя различия в статистических свойствах между обманными и чистыми изображениями. Для обнаружения обманных изображений можно использовать **анализ главных компонент** (principal component analysis, PCA) (Hendrycks, Gimpel, 2016), полагаясь на тот факт, что более поздние главные компоненты обманных изображений имеют большую дисперсию, чем у чистых изображений. **Статистический тест** (Grosse et al., 2017) использует тот факт, что распределения данных обманных и чистых изображений различны. С помощью статистического теста, такого как максимальное среднее несоответствие (Gretton et al., 2012), метод проверяет, является группа данных обманной или нет. Метод также содержит расширенную модель, которая является целевой моделью с дополнительным классом, используемым для классификации обманных входных данных.

Вспомогательные модели можно научить отличать обманные изображения от чистых. **Бинарный классификатор** (Gong et al., 2017) – это простой подход к созданию классификатора, обнаруживающего обманные изображения. Классификатор принимает изображение в качестве входных данных для классификатора и генерирует двоичную метку, которая указывает, являются входные данные обманными или нет. **Сети обнаружения противников** (Metzen et al., 2017) образуют бинарный классификатор, на вход которого поступают данные с промежуточного уровня целевой модели.

Другие методы обнаруживают обманные изображения, проверяя *согласованность* прогноза относительно входа. Это можно сделать, манипулируя входными данными или целевой моделью до приемлемого предела, а затем проверяя, соответствует ли выходной прогноз ожиданиям. Метод **KDE & BUE** (Feinman et al., 2017) решает проблему обнаружения, исследуя достоверность модели на состязательных выборках с использованием *оценок плотности ядра* (kernel density estimates, KDE) обучающих данных и *байесовских оценок неопределенности* (Bayesian uncertainty estimates, BUE) в слоях отсева. KDE вычисляются с обучающим набором в пространстве признаков последнего скрытого слоя и обнаруживают аномальные точки, которые лежат далеко от множества данных. BUE может обнаруживать обманные изображения, если распределение неопределенности, оцененное случайным отсевом, отличается от чистых данных. **Уплотнение признаков** (feature squeezing) (Xu et al., 2018) выполняет повторное квантование изображений (уменьшение глубины цвета в битах) и пространственную фильтрацию (медианную фильтрацию) входных изображений. Метод предполагает, что предсказание модели на основе чистых изображений не изменяется при применении повторного квантования или медианной фильтрации. Если после переэквантования или медианной фильтрации результат прогнозирования становится другим, метод относит входные данные к обманным изображениям.

15.11.2. Маскировка градиента

Большинство обманных атак для создания возмущений используют градиент функции потерь целевой модели. *Маскировка градиента* направлена на то, чтобы сделать информацию о градиенте целевой модели неприменимой

для создания обманных возмущений. Примерами маскировки являются *недифференцируемый градиент*, *исчезающий/взрывающийся градиент* и *стохастический градиент*.

Термометрическое кодирование (thermometer encoding) (Buckman et al., 2018) решает проблему линейной экстраполяции моделей DNN путем предварительной обработки входных данных с дискретизацией, чтобы сделать защитную модель нелинейной и *недифференцируемой*. Обычное квантование может быть неэффективным приемом защиты, поскольку возмущения также могут примерно линейно влиять на квантованные входные данные. Поэтому термометрическое кодирование учится выдавать дискретизированное значение, что позволяет модели сделать вход, эффективный для противодействия обманным возмущениям. **Трансформация изображений** (image transformation) (Guo et al., 2018) предлагает несколько подходов к обработке изображений, таких как кадрирование и масштабирование, уменьшение битовой глубины и сжатие JPEG. Также может использоваться минимизация общей дисперсии для упорядочивания каждого небольшого набора пикселей в изображении. Сшивание изображений, синтезирование изображений небольшими патчами – это еще один подход, который заменяет локальные патчи в обманном изображении чистыми патчами ближайшего соседа. Эти недифференцируемые преобразования затрудняют атакующей стороне вывод градиента функции потерь из целевой модели.

Исчезающий/взрывающийся градиент при обучении защитной модели делает градиенты функции потерь очень маленькими или большими. Такой подход отвлекает противника от атаки на входные изображения. **Защитная дистилляция** (Papernot et al., 2016) добавляет гибкости процессу классификации, поэтому модель менее уверена в своем прогнозе. Дистиллированная модель обучается прогнозировать выходные вероятности целевой модели. Эта целевая модель была обучена для достижения высокой точности с изучением более мягкого распределения вероятностей с помощью функции softmax с большой температурой $T: e^{z_n/T} / \sum_m e^{z_n/T}$. Затем дистиллированная модель используется для тестирования с небольшой T , что делает модель более надежной в прогнозировании, чем при использовании большой температуры. Это не позволяет противнику оценить градиент, поскольку градиент потерь от других классов становится близким к нулю. По методу **PixelDefend** (Song et al., 2018) генеративную сеть обучают на чистых данных, чтобы аппроксимировать их распределение, и, следовательно, побуждают сеть следовать исходному распределению, даже когда в качестве входных данных подаются обманные изображения. Эту задачу можно рассматривать как исключение состязательного воздействия. Более того, из-за большого количества параметров в генеративной сети градиенты функции потерь могут стать очень маленькими или большими, что сбивает противника с толку и не дает сделать выводы о необходимых возмущениях. **Defense-GAN** (Samanogoei et al., 2018) – это структура GAN, которая одновременно обучает генеративную сеть для имитации распределения данных и дискриминативную модель для прогнозирования того, являются ли входные данные обманными. Что касается PixelDefend, то генеративная сеть может устранять влияние

обманных возмущений. Состязательный вариационный автокодировщик (adversarial variational autoencoder, A-VAE) (Zhou et al., 2020) – это средство защиты от атак на систему распознавания лиц, которое обучает генератор на основе VAE по исходным изображениям для изучения распределения чистых данных. Во время рабочего вывода входное изображение с пониженной частотой дискретизации подается в обученный VAE, в то время как несколько скрытых кодов выбираются из кодера для создания декодированных изображений с различными деталями. Затем входные данные для модели распознавания лиц выбираются путем нахождения ближайшего соседа входных данных в декодированных изображениях. Это позволяет A-VAE выбирать входное изображение, которое, вероятно, позволит предсказать правильную метку и при этом похоже на входное изображение.

Стохастический градиент сбивает противника с толку на целевой модели для атаки, применяя рандомизированные операции к входным данным или сети. **Стохастическое отсечение** (Dhillon et al., 2018) отбрасывает случайное подмножество выходных данных скрытого слоя (или активаций) и увеличивает остальные, чтобы компенсировать это. Метод сохраняет узлы с вероятностью, пропорциональной величине их активации, и масштабирует выжившие узлы, чтобы сохранить динамический диапазон активаций в каждом слое. Этот подход может сделать предварительно обученные модели более устойчивыми к обманным изображениям без тонкой настройки. **Рандомизация** (Xie et al., 2018) использует две операции рандомизации – случайное изменение размера и заполнение – на входных изображениях, чтобы смягчить атаки злоумышленников. Хотя добавление случайного изменения размера и случайного заполнения демонстрирует небольшое падение точности на чистых изображениях, прогноз с рандомизацией демонстрирует устойчивость к атакам, ограниченным нормой.

15.11.3. Устойчивость модели

Устойчивость целевой модели к атаке со стороны противника может быть повышена за счет регуляризации, состязательного обучения или сертифицированной защиты. *Регуляризация* слоев нейронной сети делает обучение менее чувствительным к входным искажениям. *Состязательное обучение* переучивает целевую модель на обманных изображениях, созданных в результате атаки, в дополнение к обучению на чистых изображениях. Однако нет гарантии, что переученная модель будет устойчива к другим атакам, не используемым для переучивания. *Сертифицированная защита* направлена на предоставление сертификата, гарантирующего устойчивость модели в определенных пределах возмущений. Этот подход отличается от состязательного обучения и регуляризации, которые смягчают состязательные эффекты в доменах данных/признаков.

Глубокая сокращаемая сеть (deep contractive network, Gu, Rigazio, 2014) регуляризует атаки, применяя послойный штраф как потерю, которая ограничивает величину частной производной скрытых слоев в целевой модели, в дополнение к потерям, с которыми противник столкнулся при атаке. Это

позволяет сети быть менее чувствительной к изменениям входного изображения, таким как возмущения со стороны противника. **Сеть Парсевала** (Cisse et al., 2017) контролирует глобальную константу Липшица сети в рамках послойной регуляризации. Если рассматривать сеть как комбинацию функций, она может быть устойчивой к небольшим входным возмущениям за счет поддержания небольшой константы Липшица для этих функций. Сеть Парсевала решает эту проблему с помощью плотно связанных кадров Парсевала, которые определяют спектральную норму весовой матрицы сети.

Базовая модель **AdvTrain** (Goodfellow et al., 2015) переобучает целевой классификатор с помощью обманных изображений, сгенерированных FGSM, чьи метки совпадают с исходными изображениями. Следовательно, переобученный классификатор устойчив к атаке FGSM. Версия **PGD AdvTrain** (Madry et al., 2018) расширяет базовую AdvTrain, используя атаку PGD, которая создает наихудшее обманное изображение в пределах ℓ_∞ . **Ансамблевая AdvTrain** (Tramèr et al., 2018) повторно обучает целевую модель на обманных изображениях, созданных из других предварительно обученных классификаторов. Эта стратегия позволяет избежать проблемы переобучения в базовой AdvTrain, а обманные изображения из других классификаторов могут приблизиться к наихудшему обманному изображению. Ансамблевая AdvTrain более эффективна, чем предыдущие методы, поскольку процесс повторного обучения и состязательная атака разделены. **Устойчивый к атакам детектор объектов** (adversarially robust object detector, AROD) (Zhang and Wang, 2019) применяет атаку, подобную FGSM, к функциям потери классификации и локализации, которые используются при обнаружении объектов, и создает обманные изображения из каждой потери. Результирующий обманный образ – это тот, который максимизирует общую функцию потерь (как классификацию, так и локализацию). Эта же потеря затем используется для подготовки защиты. **Робастный трекер** (robust tracker, RT) (Jia et al., 2020) оценивает неизвестные обманные возмущения, созданные с учетом движения, происходящего с течением времени во входных видео, и учится устранять их влияние во время отслеживания. Метод сначала определяет потерю со стороны противника, которая включает в себя исходную метку, например ограничивающую рамку и ее класс, с целевой псевдометкой. На входе имеется изображение с неизвестными возмущениями; градиент от состязательных потерь, измеренный с использованием оценочной метки из предыдущего кадра и соответствующей ему псевдометки, вычитается из входного изображения, чтобы уменьшить эффект атаки.

Другой подход – обучение для оптимизации сертификатов, обеспечивающих надежность модели (сертифицированная защита). Исходя из модели и входных данных, верификатор выдает сертификат, если изображение гарантированно не является обманным. Это можно сделать, проверив, есть ли на расстоянии ℓ_p какое-либо изображение с меткой, отличной от исходной. Один из подходов заключается в обучении сертификата, который аппроксимирует внешние границы состязательных потерь целевой моделью. Метод **одиночной полуопределенной релаксации** (single semidefinite relaxation) генерирует сертификат, который обеспечивает верхнюю границу, где атака невозможна, учитывая целевую модель и входные данные (Raghunathan et

al., 2018). Аналогично **глубокая ReLU** (Wong, Kolter, 2018) представляет собой более глубокую сеть для обучения доказуемо устойчивых классификаторов, которые гарантированно будут устойчивы к любым ограниченным нормой обманным возмущениям в обучающем наборе. Поскольку обученный сертификат приближается к внешним границам состязательных потерь, незнакомые обманные изображения могут быть обнаружены с нулевым количеством ложноотрицательных результатов, но это не исключает ложноположительные срабатывания.

Арнаб и др. (Arnab et al., 2018) исследовали устойчивость к состязательным атакам нескольких модулей, используемых в DNN для семантической сегментации, т. е. **робастных моделей семантической сегментации** (robust semantic segmentation models, RSSM). Модели для семантической сегментации состоят из предварительно обученной модели классификации, используемой в качестве основы, и дополнительных слоев или модулей для лучшей локализации на уровне пикселей, таких как условные случайные поля, расширенные свертки, пропущенные соединения и многомасштабные сети. В ряде работ продемонстрирована атака на модели семантической сегментации с вариантами FGSM (Goodfellow et al., 2015; Kurakin et al., 2017) и показано, что модель, которая является самой точной на чистых изображениях, не обязательно оказывается самой надежной. Условные случайные поля, которые обычно используются в семантической сегментации для обеспечения соблюдения структурных ограничений, содержат вывод, который естественным образом выполняет градиентную маскировку, что приводит к повышению устойчивости к ненаправленным атакам со стороны противника. RSSM, основанный на пропущенных соединениях, например ResNet (He et al., 2016), и многомасштабная формулировка более устойчивы к атакам со стороны противника, чем модели, подобные VGG (Simonyan, Zisserman, 2014).

15.12. Выводы

В данной главе был представлен обширный обзор состязательных атак на модели машинного обучения для обработки изображений, классификации изображений, обнаружения объектов, семантической сегментации, отслеживания объектов и классификации видео, а также средств защиты от этих атак. Мы представили ключевые свойства состязательной атаки, а именно эффективность, надежность, переносимость и заметность, а также обсудили дополнительные свойства, которые необходимо учитывать при оценке атаки, такие как обнаруживаемость и обратимость. Затем мы классифицировали методы как ограниченные и неограниченные в зависимости от величины обманного возмущения, а также различали возмущения как глобальные или региональные. Мы сравнили стратегии с учетом конкретной задачи, для которой они предназначены (например, оценка движения, классификация изображений или видео, отслеживание объектов) и различных параметров белого и черного ящиков. Анализ атак позволил нам определить категории стратегий, защищающих DNN от обманных изображений. Средства защиты

могут определить, являются ли входные данные обманными, либо предотвратить атаку с помощью градиентной маскировки, где градиент исходит из функции потерь. Кроме того, мы обсудили, как сделать модели DNN более надежными с помощью состязательного обучения.

Взаимодействие между атаками и защитой станет важным направлением будущих исследований, которые помогут понять ограничения моделей DNN и разработать более надежные глубокие нейросети. Кроме того, еще одной важной областью исследования являются физические атаки, которые изменяют внешний вид реальных объектов или помещают объекты противника в окружающую среду, и соответствующие средства защиты от таких атак (Eykholt et al., 2018; Sharif et al., 2016; Brown et al., 2018; Kurakin et al., 2017; Athalye et al., 2018; Ranjan, Black, 2017; Chen et al., 2017; Thys et al., 2019; Xu et al., 2020). DNN, как правило, более уязвимы для состязательных атак, чем традиционные модели машинного обучения, поскольку сквозные обучаемые архитектуры DNN упрощают атаку, а некоторые свойства DNN еще не полностью исследованы. Преодоление уязвимости DNN к злонамеренному манипулированию изображениями и видео как непосредственно в цифровом пространстве, так и в физическом мире является ключом к их внедрению в реальных приложениях, таких как торговля, безопасность и аутентификация, а также в критических автономных системах безопасности, таких как самоуправляемые транспортные средства (Modas et al., 2020).

Благодарность

Андреа Кавалларо выражает благодарность Институту Алана Тьюринга (грант EP/N510129/1), финансируемому EPSRC, за поддержку в рамках проекта PRIMULA.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

- Allen-Zhu Z., Li Y.*, 2020. Feature purification: how adversarial training performs robust deep learning. arXiv:2005. 10190.
- Anguita D., Ghio A., Oneto L., Parra X., Reyes-Ortiz J.*, 2013. A public domain dataset for human activity recognition using smartphones. In: Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.
- Arnab A., Miksik O., Torr P. H. S.*, 2018. On the robustness of semantic segmentation models to adversarial attacks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Athalye A., Engstrom L., Ilyas A., Kwok K.*, 2018. Synthesizing robust adversarial examples. In: Proceedings of the International Conference on Machine Learning.
- Baluja S., Fischer I.*, 2018. Learning to attack: adversarial transformation networks. In: Proceedings of the AAAI Conference on Artificial Intelligence.

- Bertinetto L., Valmadre J., Henriques J. F., Vedaldi A., Torr P. H.*, 2016. Fully-convolutional Siamese networks for object tracking. In: Proceedings of the European Conference on Computer Vision.
- Bhattad A., Chong M. J., Liang K., Li B., Forsyth D.*, 2020. Unrestricted adversarial examples via semantic manipulation. In: Proceedings of the International Conference on Learning Representations.
- Biggio B., Nelson B., Laskov P.*, 2013. Poisoning attacks against support vector machines. In: Proceedings of the International Conference on Machine Learning.
- Bolton R. J., Hand D. J., et al.*, 2002. Statistical fraud detection: a review. *Statistical Science* 17, 235–255.
- Brendel W., Rauber J., Bethge M.*, 2018. Decision-based adversarial attacks: reliable attacks against black-box machine learning models. In: Proceedings of the International Conference on Learning Representations.
- Bromley J., Guyon I., LeCun Y., Säckinger E., Shah R.*, 1994. Signature verification using a Siamese time delay neural network. In: Proceedings of the Advances in Neural Information Processing Systems.
- Brown T. B., Mané D., Roy A., Abadi M., Gilmer J.*, 2018. Adversarial patch. arXiv: 1712.09665.
- Brox T., Malik J.*, 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 500–513.
- Buckman J., Roy A., Raffel C., Goodfellow I.*, 2018. Thermometer encoding: one hot way to resist adversarial examples. In: International Conference on Learning Representations.
- Carlini N., Wagner D.*, 2017. Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy.
- Carreira J., Zisserman A.*, 2017. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Chen P. Y., Zhang H., Sharma Y., Yi J., Hsieh C. J.*, 2017. Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitutemodells. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security.
- Cisse M., Bojanowski P., Grave E., Dauphin Y., Usunier N.*, 2017. Parseval networks: improving robustness to adversarial examples. In: International Conference on Machine Learning.
- Coates A., Ng A., Lee H.*, 2011. An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings.
- Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B.*, 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Cosgrove C., Yuille A.*, 2020. Adversarial examples for edge detection: they exist, and they transfer. In: Proceedings of the IEEE/CVFWinter Conference on Applications of Computer Vision.

- Das N., Shanbhogue M., Chen S., Hohman F., Chen L., Kounavis M. E., Chau D. H., 2017. Keeping the bad guys out: protecting and vaccinating deep learning with JPEG compression. arXiv:1705.02900.
- Deng J., Dong W., Socher R., Li L. J., Li K., Fei-Fei L., 2009. Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Dhillon G. S., Azizzadenesheli K., Lipton Z. C., Bernstein J. D., Kossaiji J., Khanna A., Anandkumar A., 2018. Stochastic activation pruning for robust adversarial defense. In: International Conference on Learning Representations.
- Dong Y., Pang T., Su H., Zhu J., 2019. Evading defenses to transferable adversarial examples by translationinvariant attacks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Dosovitskiy A., Fischer P., Ilg E., Häusser P., Hazirbas C., Golkov V., Smagt P.v.d., Cremers, D. Brox T., 2015. FlowNet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision.
- Dziugaite G. K., Ghahramani Z., Roy D. M., 2016. A study of the effect of JPG compression on adversarial images. arXiv:1608.00853.
- Engstrom L., Ilyas A., Santurkar S., Tsipras D., Tran B., Madry A., 2019. Adversarial robustness as a prior for learned representations. arXiv:1906.00945.
- Everingham M., Eslami S. A., Van Gool L., Williams C. K., Winn J., Zisserman A., 2015. The Pascal visual object classes challenge: a retrospective. International Journal of Computer Vision 111, 98–136.
- Eykholt K., Evtimov I., Fernandes E., Li B., Rahmati A., Xiao C., Prakash A., Kohno T., Song D., 2018. Robust physical-world attacks on deep learning visual classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Farbman Z., Fattal R., Lischinski D., Szeliski R., 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. ACM Transactions on Graphics 27, 1–10.
- Feinman R., Curtin R. R., Shintre S., Gardner A. B., 2017. Detecting adversarial samples from artifacts. arXiv:1703.00410.
- Fischer V., Kumar M. C., Metzen J. H., Brox T., 2017. Adversarial examples for semantic image segmentation. In: Proceedings of the International Conference on Machine Learning Workshop.
- Geiger A., Lenz P., Urtasun R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the Conference on Computer Vision and Pattern Recognition.
- Gong Z., Wang W., Ku W. S., 2017. Adversarial and clean data are not twins. arXiv:1704.04960.
- Goodfellow I., Shlens J., Szegedy C., 2015. Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations.
- Goodfellow I. J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014. Generative adversarial nets. In: Proceedings of the International Conference on Neural Information Processing Systems.
- Gretton A., Borgwardt K. M., Rasch M. J., Schölkopf B., Smola A., 2012. A kernel two-sample test. The Journal of Machine Learning Research 13, 723–773.

- Grosse K., Manoharan P., Papernot N., Backes M., McDaniel P., 2017. On the (statistical) detection of adversarial examples. arXiv:1702.06280.
- Gu S., Rigazio L., 2014. Towards deep neural network architectures robust to adversarial examples. arXiv preprint. arXiv:1412.5068.
- Guo C., Gardner J., You Y., Wilson A. G., Weinberge K., 2019. Simple black-box adversarial attacks. In: Proceedings of the International Conference on Machine Learning.
- Guo C., Rana M., Cisse M., van der Maaten L., 2018. Countering adversarial images using input transformations. In: Proceedings of the International Conference on Learning Representations.
- Hara K., Kataoka H., Satoh Y., 2018. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- He K., Zhang X., Ren S., Sun J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- Hendrycks D., Gimpel K., 2016. Early methods for detecting adversarial images. arXiv:1608.00530.
- Hosseini H., Poovendran R., 2018. Semantic adversarial examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.
- Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.
- Huang G., Liu Z., Van Der Maaten L., Weinberger K. Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Huang G. B., Mattar M., Berg T., Learned-Miller E., 2008. Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition.
- Ilg E., Mayer N., Saikia T., Keuper M., Dosovitskiy A., Brox T., 2017. FlowNet 2.0: evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Ilyas A., Engstrom L., Athalye A., Lin J., 2018. Black-box adversarial attacks with limited queries and information. In: Proceedings of the International Conference on Machine Learning.
- Isola P., Zhu J., Zhou T., Efros A. A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Janai J., Guney F., Ranjan A., Black M., Geiger A., 2018. Unsupervised learning of multi-frame optical flow with occlusions. In: Proceedings of the European Conference on Computer Vision.
- Jia S., Ma C., Song Y., Yang X., 2020. Robust tracking against adversarial attacks. In: Proceedings of the European Conference on Computer Vision.
- Jiang L., Ma X., Chen S., Bailey J., Jiang Y., 2019. Black-box adversarial attacks on video recognition models. In: Proceedings of the ACM International Conference on Multimedia.

- Jiang Y., Wu Z., Wang J., Xue X., Chang S., 2018. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 352–364.
- Kay W., Carreira J., Simonyan K., Zhang B., Hillier C., Vijayanarasimhan S., Viola F., Green T., Back T., Natsev P., Suleyman M., Zisserman A., 2017. The kinetics human action video dataset. *arXiv:1705.06950 [cs.CV]*.
- Krizhevsky A., Hinton G., et al., 2009. Learning multiple layers of features from tiny images.
- Krizhevsky A., Sutskever I., Hinton G. E., 2012. ImageNet classification with deep convolutional neural networks. In: *Proceedings of the Advances in Neural Information Processing Systems*.
- Kuehne H., Jhuang H., Garrote E., Poggio T., Serre T., 2011. HMDB: a large video database for human motion recognition. In: *Proceedings of the International Conference on Computer Vision*. Stockholm, Sweden.
- Kurakin A., Goodfellow I., Bengio S., 2017a. Adversarial examples in the physical world. In: *Proceedings of the International Conference on Learning Representations – Workshops*. Toulon, France.
- Kurakin A., Goodfellow I., Bengio S., 2017b. Adversarial machine learning at scale. In: *Proceedings of the International Conference on Learning Representations*.
- LeCun Y., 1998. The MNIST database of handwritten digits.
- LeCun Y., Bottou L., Bengio Y., Haffner P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- Li B., Wu W., Wang Q., Zhang F., Xing J., Yan J., 2019. SiamRPN++: evolution of Siamese visual tracking with very deep networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Li B., Yan J., Wu W., Zhu Z., Hu X., 2018. High performance visual tracking with Siamese region proposal network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Li C. Y., Sanchez-Matilla R., Shahin Shamsabadi A., Mazzon R., Cavallaro A., 2021. On the reversibility of adversarial attacks.
- Li C. Y., Shahin Shamsabadi A., Sanchez-Matilla R., Mazzon R., Cavallaro A., 2019a. Scene privacy protection. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.
- Li S., Neupane A., Paul S., Song C., Krishnamurthy S. V., Chowdhury A. K. R., Swami A., 2019b. Stealthy adversarial perturbations against real-time video classification systems. In: *Proceedings of the Network and Distributed Systems Security Symposium*.
- Liang S., Wei X., Yao S., Cao X., 2020. Efficient adversarial attacks for visual object tracking. In: *Proceedings of the European Conference on Computer Vision*.
- Lin T. Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C. L., 2014. Microsoft COCO: common objects in context. In: *Proceedings of the European Conference on Computer Vision*.
- Lo S., Patel V. M., 2020. MultAV: multiplicative adversarial videos. *arXiv:2009.08058*.
- Long J., Shelhamer E., Darrell T., 2015. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Lu J., Sibai H., Fabry E., 2017. Adversarial examples that fool detectors. arXiv: 1712.02494 [cs.CV].
- Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A., 2018. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the International Conference on Learning Representations.
- Martin D. R., Fowlkes C. C., Malik J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 530–549.
- Materzynska J., Berger G., Bax I., Memisevic R., 2019. The jester dataset: a large-scale video dataset of human gestures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.
- Metzen J. H., Genewein T., Fischer V., Bischoff B., 2017. On detecting adversarial perturbations. arXiv:1702.04267.
- Meyer T. A., Whateley B., 2004. SpamBayes: effective open-source, Bayesian based, email classification system. In: Proceedings of the Conference on Email and Anti-Spam.
- Modas A., Moosavi-Dezfooli S. M., Frossard P., 2019. Sparsefool: a few pixels make a big difference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Modas A., Sanchez-Matilla R., Frossard P., Cavallaro A., 2020. Towards robust sensing for autonomous vehicles: an adversarial perspective. *IEEE Signal Processing and Magazine* 37, 14–23.
- Moosavi-Dezfooli S. M., Fawzi A., Fawzi O., Frossard P., 2017. Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Moosavi-Dezfooli S. M., Fawzi A., Frossard P., 2016. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Mopuri K. R., Garg U., Babu R. V., 2017. Fast feature fool: a data independent approach to universal adversarial perturbations. In: Proceedings of the British Machine Vision Conference.
- Mopuri K. R., Ojha U., Garg U., Babu R. V., 2018. NAG: network for adversary generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Mueller M., Smith N., Ghanem B., 2016. A benchmark and simulator for UAV tracking. In: Proceedings of the European Conference on Computer Vision.
- Narodytska N., Kasiviswanathan S. P., 2017. Simple black-box adversarial attacks on deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.
- Netzer Y., Wang T., Coates A., Bissacco A., Wu B., Ng A. Y., 2011. Reading digits in natural images with unsupervised feature learning. In: Workshop on Deep Learning and Unsupervised Feature Learning (in conjunction with Neural Information Processing Systems).
- Ng Y. H. J., Hausknecht M., Vijayanarasimhan S., Vinyals O., Monga R., Toderici G., 2015. Beyond short snippets: deep networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

- Papernot N., McDaniel P., Goodfellow I.*, 2016a. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv:1605.07277.
- Papernot N., McDaniel P., Jha S., Fredrikson M., Celik Z. B., Swami A.*, 2016b. The limitations of deep learning in adversarial settings. In: Proceedings of the IEEE European Symposium on Security and Privacy.
- Papernot N., McDaniel P., Wu X., Jha S., Swami A.*, 2016c. Distillation as a defense to adversarial perturbations against deep neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy.
- Poursaeed O., Katsman I., Gao B., Belongie S.*, 2018. Generative adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Raghunathan A., Steinhardt J., Liang P.*, 2018. Certified defenses against adversarial examples. arXiv:1801.09344.
- Ranjan A., Black M. J.*, 2017. Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Ranjan A., Janai J., Geiger A., Black M. J.*, 2019. Attacking optical flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Redmon J., Farhadi A.*, 2017. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Ren S., He K., Girshick R., Sun J.*, 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 1137–1149.
- Revaud J., Weinzaepfel P., Harchaoui Z., Schmid C.*, 2015. Epicflow: edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Salimans T., Ho J., Chen X., Sidor S., Sutskever I.*, 2017. Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864.
- Samangouei P., Kabkab M., Chellappa R.*, 2018. Defense-GAN: protecting classifiers against adversarial attacks using generative models. In: Proceedings of the International Conference on Learning Representations.
- Sanchez-Matilla R., Li C. Y., Shamsabadi A. S., Mazzon R., Cavallaro A.*, 2020. Exploiting vulnerabilities of deep neural networks for privacy protection. IEEE Transactions on Multimedia 22, 1862–1873.
- Santurkar S., Ilyas A., Tsipras D., Engstrom L., Tran B., Madry A.*, 2019. Image synthesis with a single (robust) classifier. In: Proceedings of the Conference on Neural Information Processing Systems.
- Shamsabadi A. S., Oh C., Cavallaro A.*, 2020a. Edgefool: an adversarial image enhancement filter. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.
- Shamsabadi A. S., Oh C., Cavallaro A.*, 2020b. Semantically adversarial learnable filters. arXiv:2008.06069.
- Shamsabadi A. S., Sanchez-Matilla R., Cavallaro A.*, 2020c. ColorFool: semantic adversarial colorization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

- Sharif M., Bhagavatula S., Bauer L., Reiter M. K.*, 2016. Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.
- Shi Y., Wang S., Han Y.*, 2019. Curls & whey: boosting black-box adversarial attacks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Simonyan K., Zisserman A.*, 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
- Song Y., Kim T., Nowozin S., Ermon S., Kushman N.*, 2018. Pixeldefend: leveraging generative models to understand and defend against adversarial examples. In: Proceedings of the International Conference on Learning Representations.
- Soomro K., Zamir A. R., Shah M.*, 2012. UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402 [cs.CV].
- Su J., Vargas D. V., Sakurai K.*, 2019. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23, 828–841.
- Sun D., Yang X., Liu M., Kautz J.*, 2018. PWC-net: CNNs for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Szegedy C., Ioffe S., Vanhoucke V., Alemi A. A.*, 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.*, 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z.*, 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Szegedy C., Zaremba W., Sutskever I., Bruna J., Erhan D., Goodfellow I., Fergus R.*, 2014. Intriguing properties of neural networks. In: Proceedings of the International Conference on Learning Representations.
- Thys S., Van Ranst W., Goedeme T.*, 2019. Fooling automated surveillance cameras: adversarial patches to attack person detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
- Tramèr F., Kurakin A., Papernot N., Goodfellow I., Boneh D., McDaniel P.*, 2018. Ensemble adversarial training: attacks and defenses. In: Proceedings of the International Conference on Learning Representations.
- Tran D., Bourdev L., Fergus R., Torresani L., Paluri M.*, 2015. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision.
- Tsipras D., Santurkar S., Engstrom L., Turner A., Madry A.*, 2019. Robustness may be at odds with accuracy. In: Proceedings of the International Conference on Representation Learning.
- Tu C. C., Ting P., Chen P. Y., Liu S., Zhang H., Yi J., Hsieh C. J., Cheng S. M.*, 2019. Autozoom: autoencoder-based zeroth order optimization method for attacking black-box neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence.

- Wei X., Liang S., Chen N., Cao X., 2019. Transferable adversarial attacks for image and video object detection. In: Proceedings of the International Joint Conference on Artificial Intelligence.
- Wong E., Kolter Z., 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In: Proceedings of the International Conference on Machine Learning.
- Wu Y., Lim J., Yang M., 2015. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 1834–1848.
- Xiao C., Li B., Zhu J., He W., Liu M., Song D., 2018. Generating adversarial examples with adversarial networks. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence.
- Xiao H., Rasul K., Vollgraf R., 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747.
- Xie C., Wang J., Zhang Z., Ren Z., Yuille A., 2018. Mitigating adversarial effects through randomization. In: Proceedings of the International Conference on Learning Representations.
- Xie C., Wang J., Zhang Z., Zhou Y., Xie L., Yuille A., 2017. Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE International Conference on Computer Vision.
- Xie C., Zhang Z., Zhou Y., Bai S., Wang J., Ren Z., Yuille A. L., 2019. Improving transferability of adversarial examples with input diversity. In: Proceedings of the Computer Vision and Pattern Recognition.
- Xie S., Tu Z., 2015. Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision.
- Xu K., Zhang G., Liu S., Fan Q., Sun M., Chen H., Chen P., Wang Y., Lin X., 2020. Adversarial t-shirt! Evading person detectors in a physical world. In: Proceedings of the European Conference on Computer Vision.
- Xu W., Evans D., Qi Y., 2018. Feature squeezing: detecting adversarial examples in deep neural networks. In: Proceedings of the Network and Distributed System Security Symposium.
- Yu F., Koltun V., Funkhouser T., 2017. Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Zagoruyko S., Komodakis N., 2016. Wide residual networks. In: Proceedings of the British Machine Vision Conference.
- Zajac M., Żoła K., Rostamzadeh N., Pinheiro P. O., 2019. Adversarial framing for image and video classification. In: Proceedings of the AAAI Conference on Artificial Intelligence.
- Zhang H., Wang J., 2019. Towards adversarially robust object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Zhang Z., Peng H., 2019. Deeper and wider Siamese networks for real-time visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Zhou B., Lapedriza A., Khosla A., Oliva A., Torralba A., 2017. Places: a 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, 1452–1464.
- Zhou J., Liang C., Chen J., 2020. Manifold projection for adversarial defense on face recognition. In: Proceedings of the European Conference on Computer Vision.

ОБ АВТОРАХ ГЛАВЫ

Чанги О – преподаватель Школы электроники и компьютерных наук и Центра интеллектуального восприятия Лондонского университета королевы Марии, Великобритания. Он получил степень бакалавра, магистра и доктора наук в области электротехники и электроники в Университете Йонсей, Сеул, Южная Корея, в 2011, 2013 и 2018 гг. соответственно. С 2018 по 2019 г. был научным сотрудником в докторантуре Лондонского университета королевы Марии, Великобритания.

Алессио Зомперо – научный сотрудник с докторской степенью в области мультимодального восприятия в Школе электронных технологий и компьютерных наук и Центре интеллектуального восприятия Лондонского университета королевы Марии, Великобритания. Получил степень магистра в области телекоммуникаций в Университете Тренто, Италия, в 2015 г. и докторскую степень в области электронных технологий в Лондонском университете королевы Марии, Великобритания, в 2020 г.

Андреа Кавалларо – профессор в области обработки мультимедийных сигналов в Лондонском университете королевы Марии (QMUL) и научный сотрудник Института Алана Тьюринга, Британского национального института науки о данных и искусственного интеллекта. Является членом Международной ассоциации распознавания образов; директор Центра интеллектуального восприятия QMUL; был главным редактором журнала *Signal Processing: Image Communication*; старшим редактором раздела *IEEE Transactions on Image Processing*; председателем технического комитета IEEE по обработке изображений, видео и многомерных сигналов и заслуженным лектором Общества обработки сигналов IEEE.

Предметный указатель

A

action/behavior recognition, 24
attention map, 25

C

cascaded detection and regression, 22
CNN, convolutional neural network, 21

D

data augmentation, 22
deep generative model, 23
DNN, deep neural network, 21

F

face recognition, 23
face synthesis, 23

K

knowledge distillation, 23

L

loss function, 22
low-rank factorization, 23

M

model decay, 24
multiscale feature representations, 22
multitask losses, 22

P

parameter pruning, 23

R

RANSAC, 51
region-of-interest pooling, 22
region proposal networks, 22

S

Siamese tracker, 24
sparse representation, 23
style transfer, 23

A

Автокодировщик, 262
 вариационный, 263
Адаптация домена, 315, 358
 без учителя, 317, 354
 с частичным привлечением
 учителя, 354
Активное видение, 478
Алгоритм
 8-точечный, 88
 ближайшего соседа, 28
 максимизации ожидания, 94, 260
 полуквадратичного разделения, 608
Анализ главных компонент, 290
Аномалия расхождения
Кульбака–Лейблера, 588
Апостериорный максимум, 606
Атака
 без вывода, 643
 белого ящика, 643
 границная, 654
 заметность, 644
 замещающего черного ящика, 660
 методом случайного поиска, 661

направленная, 641
 ненаправленная, 641
 обнаруживаемость, 644
 обратимость, 644
 оптического потока, 654
 переносимость, 643
 путем оценки градиента, 660
 робастность, 643
 с выводом метки, 643
 с распределением-выводом, 643
 черного ящика, 643
 эффективность, 643
 Атрибуты классов, 497
 Аффорданс, 476

Б

Бинокулярное зрение, 61

В

Вариационный автокодировщик, 571
 Вектор мягкого присвоения, 406
 Векторная символическая архитектура, 507
 Вергенция, 61
 Видимая область, 144
 Визуально-языковая генерация, 23
 Визуальный паттерн, 254
 атомарный, 256
 Внутрикластерная сумма квадратов, 221
 Возмущения
 глобальные, 644
 неограниченные, 645
 области, 644
 ограниченные, 645
 Восстановление изображения, 605
 сверхразрешение, 605
 удаление размытия, 605
 шумоподавление, 605
 Выборка ближайшего соседа, 157
 Выравнивание ракурса, 323

Г

Гауссова модель смещения, 576
 Гауссов шум, 45

Генеративное воспроизведение, 381
 Геометрический самоансамбль, 612
 Гильбертово пространство
 воспроизводящего ядра, 317
 Гиперграф, 441
 Гипернимия слов, 488
 Гиперпараметр
 глубина, 293
 паддинг, 293
 страйд, 293
 Гипотеза лотерейного билета, 219
 Гистограмма ориентированных
 градиентов, 139, 406, 578
 Глубинная регрессия, 372
 Глубокая генеративная модель, 23
 Грамматика контекстно-
 независимая, 490
 Грамматика действий, 489
 Граф соответствия, 55

Д

Двойственность, 395
 Действие, 475, 516
 Декодер, 262
 Дерево деятельности, 489
 Детектор
 Виолы–Джонса, 138
 многомасштабный, 154
 объектов
 двухэтапный, 139
 одноэтапный, 139
 углов, 40
 Деятельность, 475, 516
 Дискретное преобразование
 Фурье, 438
 Дискриминация с обучением, 367
 Дистилляция знаний, 23, 378
 Дистрактор, 391
 Долгая краткосрочная память, 521
 Дополнение данных, 22, 296
 Дополненная реальность, 420

З

Задача прогнозирования взгляда, 549
 Замораживание параметров, 380

Знания

- дистилляция, 216, 225
- скрытые, 216

- Значимые факторы, 260

- Зрение многоакурсное, 83

- Зрительная система, 84

- Зрительное восприятие, 254

И

- Извлечение готовых признаков, 296

- Инвариантность, 452

- Индекс Жаккара, 396

- Индуктивная предпосылка, 257

- Инициализатор модели, 465

- Инициализация весов, 381

К

- Калибровка камеры, 77

- внешняя, 80

- внутренняя, 80

- Карта

- внимания, 25

- интенсивности, 461

- признаков, 293

- Каскадное обнаружение, 22

- Катастрофическое забывание, 355

- Качество предсказания, 527

- Квантование

- без данных, 224

- весов, 23, 216

- линейное/равномерное, 222

- после настройки, 223

- сети, 221

- Кластеризация k-средних, 221

- Клика, 55

- Когнитивный диалог, 488

- Кодировщик, 262

- Коника, 77

- Контакт, 489

- Косинусное подобие, 301

- Коэффициент

- забывания, 402

- расширения, 234

- Критерий

- значимости, 219

- остановки, 330

М

- Максимальное среднее

- расхождение, 337

- Марковские случайные поля, 258

- Маскировка градиента, 668

- Масштабирование

- укрупняющее (апскейлинг), 462

- уменьшающее (даунскейлинг), 462

- Масштабное пространство, 406

- Матрица

- совпадений, 100

- существенная, 84

- фундаментальная, 84

- Машина опорных векторов, 575

- Метаклассы действий, 490

- Метаобновитель, 466

- Метаобучение, 465

- Метод

- аппроксимации границы

- решения, 660

- бисекции диаметра, 47

- глубокой развертки, 607

- жестких шаблонов, 59

- локальных бинарных шаблонов, 290

- локтя, 549

- максимальной клики, 55

- машины опорных векторов, 107

- оптимальных направлений, 320

- пакета признаков, 59

- переменного направления

- множителей, 607

- площади под кривой, 549

- полуквадратичного разделения, 607

- последовательной повторной

- выборки коэффициентов, 585

- стохастического градиентного

- спуска, 149

- фильтрации Габора, 291

- хорд и касательных, 47

- эпиполярной линии, 62

- Минимизация эмпирического

- риска, 431

- Мир

- закрытый, 392

- открытый, 392

- Мировая точка, 77

Многозадачные потери, 22
 Многомасштабное представление признаков, 22

Модель

- бустинг, 96
- Гаусса смешанная, 94
- генеративная, 571
- деградации одиночного изображения, 606
- деградации
 - с шумоподавлением, 606
- деформируемых частей, 59, 139
- дискриминативная, 571
- коммутирующая, 584
- наблюдения, 571
- развертывание, 392
- скрытая марковская, 572
- события, 516, 518
- текстуры, 258
- точная настройка, 23
- устаревание, 434
- фильтра Калмана, 572

Морфинг атрибутов, 282

Н

Накопление доказательств, 50
 Настройка, 221

- с учетом квантования, 224

 Нейронная сеть

- байесовская, 571
- восстановления изображений
 - с глубокой разверткой, 620
- генеративно-сопоставительная, 264, 571
- глубокая, 21, 108
- графовая сверточная, 504
- деконволюции, 115
- древовидная CNN, 441
- искусственная, 106
- пирамиды признаков, 156
- предсказания области, 464
- прогнозирования регионов, 147
- распознавания устаревания, 466
- сверточная, 21, 108
- сверточная без обучения, 440
- сиамская, 121

- с троичными весами, 223
- структурозависимая, 440

Низкоранговая факторизация, 216, 220
 Нормализованная
 кросс-корреляция, 398

О

Область наблюдения, 464
 Обманное изображение, 640
 Обнаружение

- краев, 31
- объектов, 137

 Обобщенная ошибка, 569
 Обобщенное состояние, 568
 Обратное распространение ошибки, 108
 Обучение

- бесконечное, 506
- непрерывное, 373
- однократное, 391
- слабое, 354
- совмещенное, 375
- с переносом, 315
- с учителем, 108
- трансдуктивное трансферное, 358

 Обучение без ознакомления, 497
 Объект

- визуальный сопоставительный, 26
- действия, 24
- деятельность, 24
- характеристики, 24

 Ограничение разреженности, 220
 Однородность кластеризации, 552
 Однородные координаты, 78
 Окклюзия, 45
 Окно

- поиска, 401
- Ханна (косинусное), 401

 Оператор

- Боде, 41
- Гессе, 41
- Кэнни, 33
- Лапласа, 41
- Собеля, 32
- Харриса, 42

Оптимизатор модели, 465
 Оптимизация чувствительности
 обнаружения, 35
 Оптический поток, 582
 Остаточная ошибка, 323
 Отношение
 коэффициентов, 69
 сигнал–шум, 35
 Отрицательный перенос, 368
 Отсечение коэффициентов, 23
 Отслеживание объекта, 390
 визуальное, 390
 длительное, 431
 инкрементное, 437
 краткосрочное, 431
 через обнаружение, 391
 Оценка
 обособленности, 396
 Оценка плотности ядра, 576
 Ошибка
 перцептивного предсказания, 518,
 525
 предсказания, 520
 реконструкции, 589
 средняя угловая, 550

П

Паддинг, 109
 Парадокс качественного
 обнаружения, 149
 Параметры камеры
 внешние, 82
 внутренние, 82
 Партономия, 516
 Передача внимания, 225
 Перекрестная энтропия, 161, 309
 Перенос
 обучения, 295
 стиля, 23, 267
 Пересечение над объединением, 118
 Перетасовка каналов, 235
 Перспектива
 полная, 64
 слабая, 64
 n-точечная, 64
 Перспективная проекция, 60

Перцептивная обработка, 519
 Пирамида
 изображений, 152
 масштаба, 407
 Показатель объектности, 148
 Поле восприятия (рецептивное), 109
 Полилинейная алгебра, 318
 Попарная классификация, 300
 Порядок тензора, 319
 Потеря фокальная (очаговая), 161
 Правило равных площадей, 40
 Преобразование
 перспективное, 69
 Хафа, 46
 обобщенное, 48
 Привязка, 22, 148
 Признаки Хаара, 139
 Приор, 261
 Прогрессивное улучшение, 151
 Проекция
 перспективная слабая, 64
 полноперспективная, 64
 Прореживание
 гранулярность, 217
 итеративное, 220
 крупномодульное, 218
 мелкомодульное, 217
 на ходу, 220
 параметров, 216
 структурированное, 218
 Пространственное внимание, 535
 Пространство параметров, 46
 Прямое унитарное кодирование, 497
 Пулинг, 108
 пирамидальный, 143

Р

Разделенное представление, 257
 Разметка слабая, 354
 Разреженное представление, 23
 Распад модели, 24
 Распознавание
 действий и поведения, 24
 лиц, 23, 289
 Расстояние Вассерштейна, 337
 Регуляризация, 606

Рецептивное поле нейрона, 293
 Робастная статистика, 45
 Робастность, 45

С

Самосознание, 569
 Свертка
 глубинная, 232
 с разделением по глубине, 233
 точечная групповая, 235
 Сверхразрешение одиночного изображения, 606
 Сверхсостояние, 581
 Свойство проективности, 47
 Сегментация
 видеообъектов, 416
 семантическая, 116, 354
 событий, 24
 экземпляров видео, 417
 Семантический разрыв, 496
 Семантическое пространство представления, 497
 Сенсор
 проприоцептивный, 570
 экстероцептивный, 570
 Сиамский трекер, 24
 Сила угла, 41
 Символьное знание, 24
 Синономия слов, 488
 Синтез лица, 23
 Система саморазвивающаяся, 25
 Слияние видимых областей, 22
 Сложное отношение, 70
 Собственное
 движение, 73
 значение, 103, 290
 Собственные отображения матрицы Лапласа, 317
 Собственный
 вектор, 103, 290
 фильтр, 103
 Событие, 516
 Стохастический градиент, 670
 Страйд, 109
 Структурированный случайный лес, 481

Субактивность, 484
 Сфера Гаусса, 74
 Схема событий, 520

Т

Тексель, 99
 Текстон, 268
 Теория
 воплощенного познания, 475
 сегментации событий, 518
 Тесселяция, 104
 Точка схода, 73
 Трансферное обучение. См. *Перенос обучения*
 Трекер
 глубокий, 438
 неглубокий, 436
 сиамский, 448
 Треклет, 124
 Триплетные потери, 302
 Трудные отрицательные образцы, 160

У

Умножение-накопление, 226
 Управляемый рекуррентный блок, 524
 Уравнение общей точки, 60
 Ускоритель
 граничный, 229
 облачный, 229
 Условная генерация изображений, 257
 Усредненное лицо, 290

Ф

Факторизация низкого ранга, 23
 Фильтр
 дискриминативный
 корреляционный, 403
 Калмана, 573
 с нулевым влиянием, 581
 частиц, 573
 чисто фазовый, 395
 Фоновые метки, 376

Функция

- потерь, 22
 - точечной экстраполяции, 49
- Функция потерь, взвешенная по движению, 538

Ц

- Центроидный профиль, 43

Э

- Эквивариантность, 452
- вращения, 458
 - масштаба, 462
 - переноса, 455
- Энергия текстуры, 101
- Эпиполюс, 84
- Эпиполярная плоскость, 84

Книги издательства «ДМК ПРЕСС»
можно купить оптом и в розницу
в книготорговой компании «Галактика»
(представляет интересы издательств
«ДМК ПРЕСС», «СОЛОН ПРЕСС», «КТК Галактика»).

Адрес: г. Москва, пр. Андропова, 38, оф. 10;
тел.: **(499) 782-38-89**, электронная почта: **books@aliants-kniga.ru**.

При оформлении заказа следует указать адрес (полностью),
по которому должны быть высланы книги;
фамилию, имя и отчество получателя.

Желательно также указать свой телефон и электронный адрес.

Эти книги вы можете заказать и в интернет-магазине: **<http://www.galaktika-dmk.com/>**.

Редакторы издания Рой Дэвис, Мэтью Терк

Компьютерное зрение. Современные методы и перспективы развития

Главный редактор	<i>Мовчан Д. А.</i>
	<i>dmkpress@gmail.com</i>
Зам. главного редактора	<i>Сенченкова Е. А.</i>
Перевод	<i>Яценков В. С.</i>
Корректор	<i>Синяева Г. И.</i>
Верстка	<i>Чаннова А. А.</i>
Дизайн обложки	<i>Мовчан А. Г.</i>

Гарнитура PT Serif. Печать цифровая.
Усл. печ. л. 56,06. Тираж 200 экз.

Веб-сайт издательства: **www.dmkpress.com**

Книга рассказывает о передовых методах компьютерного зрения. Представлены четкие объяснения принципов и алгоритмов, на которых оно основано; особое внимание уделяется методам глубокого обучения. Все ключевые принципы проиллюстрированы примерами реального применения.

Издание адресовано исследователям и практикам в области передовых методов компьютерного зрения, а также тем, кто изучает эту технологию самостоятельно или в рамках вузовского курса.

В числе рассматриваемых тем:

- генеративные состязательные сети;
- обучение с подкреплением и самообучение;
- извлечение надежных признаков;
- обнаружение и визуальное сопровождение объектов;
- семантическая сегментация;
- лингвистические описания изображений;
- визуальный поиск 3D-форм;
- обнаружение аномалий.

О редакторах:

Рой Дэвис – почетный профессор факультета машинного зрения в университете Лондона, почетный член Британской ассоциации машинного зрения и член Международной ассоциации распознавания образов. Работает над многими аспектами зрения – от обнаружения признаков и подавления шума до робастного сопоставления образов и реализации практических задач зрения в реальном времени.

Мэтью Тёрк – президент Технологического института Toyota в Чикаго и почетный профессор Калифорнийского университета в Санта-Барбаре. Обладатель нескольких наград Международной ассоциации распознавания образов и Ассоциации вычислительной техники за вклад в компьютерное зрение, распознавание лиц и мультимодальное взаимодействие.

Интернет-магазин:
www.dmkpress.com

Оптовая продажа:
КТК «Галактика»
books@aliens-kniga.ru



ISBN 978-5-93700-148-1



9 785937 001481 >