

Санкт-Петербургский государственный университет

Кафедра информатики

Группа 23.Б16-мм

Распознавание дорожной разметки парковочных мест по fisheye камерам

ГРУДИНИН Михаил Артемович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
старший преподаватель кафедры системного программирования, к. т. н., Литвинов Ю. В.

Консультант:
инженер-программист АО «Кама» Осечкина М. С.

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор	5
2.1. Классические методы	5
2.2. Нейросетевые методы	7
2.3. Результаты обзора	14
3. Описание решения	15
4. Эксперимент	17
4.1. Исследовательские вопросы	17
4.2. Настройки эксперимента	17
4.3. Результаты эксперимента	19
Заключение	20
Список литературы	21

Введение

Распознавание дорожной разметки парковочных мест с использованием камер является ключевым аспектом современных технологий автоматизации парковки. В условиях растущего числа автомобилей и ограниченного пространства для парковки, автоматические системы становятся неотъемлемой частью городской инфраструктуры, обеспечивая повышение безопасности и комфорта для водителей.

В настоящее время системы компьютерного зрения активно внедряются в автомобильную индустрию, позволяя реализовать множество полезных функций. Традиционно для этих целей используются камеры стандартного типа, такие как камера-обскура (pinhole). Однако с развитием технологий, все большее внимание привлекают камеры кругового обзора, известные как «рыбий глаз» (fisheye). Такие камеры способны охватывать значительно больший угол обзора благодаря характерным искажениям изображения.

Использование fisheye камер позволяет сократить количество устройств, необходимых для полного контроля окружающего пространства, что делает системы более эффективными и экономически выгодными. Снижение затрат на оборудование и установку является важным преимуществом, особенно при массовом производстве и внедрении подобных технологий. Такие камеры всё чаще устанавливаются на современные автомобили, что требует адаптации существующих алгоритмов компьютерного зрения для их корректной работы в условиях искажений fisheye.

Проблема, с которой сталкиваются существующие системы, заключается в том, что большинство решений, разрабатываемых для камер типа pinhole, плохо адаптируются под fisheye камеры из-за геометрических искажений изображения. Следовательно, необходимо исследовать и адаптировать существующие решения распознавания дорожной разметки парковочных мест, чтобы они эффективно работали в условиях искажений, характерных для fisheye.

1. Постановка задачи

Целью работы является разработка системы для распознавания дорожной разметки парковочных мест на изображениях, полученных с fisheye камер автомобиля.

Для её выполнения были поставлены следующие задачи:

1. Провести обзор существующих решений распознавания дорожной разметки парковочного места.
2. Реализовать модуль распознавания на основе лучшего метода.
3. Оценить эффективность разработанной системы на тестовых данных и провести анализ результатов.

2. Обзор

Целью данного обзора является анализ существующих решений и технологий в области распознавания дорожной разметки парковочных мест с использованием fisheye камер. Основная задача – выделение сильных и слабых сторон различных подходов, выбор оптимальных технологий для реализации модуля.

Для подготовки обзора изучались научные и технические публикации, посвящённые методам компьютерного зрения, обработке изображений и решению задачи распознавания дорожной разметки парковочных мест. Основными источниками информации стали такие платформы, как arXiv, Google Scholar, Medium, eLibrary и КиберЛенинка. Дополнительно изучалась техническая документация и специализированные статьи, опубликованные в профильных изданиях. Прямых решений, которые выполняли бы сегментацию дорожной разметки парковочных мест по fisheye изображениям, найти не удалось. Однако существуют подходы, которые могут быть адаптированы для этой задачи. В обзоре рассмотрены два основных подхода: классический и нейросетевой.

2.1. Классические методы

Классические методы обработки изображений предполагают использование детерминированных алгоритмов, таких как выделение границ, фильтрация и сегментация по цвету. Рассмотрим примеры их применения для распознавания дорожной разметки.

2.1.1. Сегментация с использованием кластеризации

Одним из методов, который решает задачу сегментации, является метод кластеризации К-средних, подробно рассмотренный в работе [7]. Это метод, основанный на разделении данных на K групп, которые имеют схожие характеристики. Этот подход используется для обработки изображений, где цель — разделить изображение на несколько областей, каждая из которых соответствует определенной категории.

Основные этапы метода К-средних для сегментации:

- **Преобразование изображения в подходящий формат для кластеризации.** Обычно изображения преобразуют в векторы признаков (например, RGB-значения, интенсивности пикселей, цветовые пространства и другие признаки). Кластеризация на изображении использует центры кластеров, которые характеризуются цветом, расположением и яркостью.
- **Задание количества кластеров K .** В К-средних задается количество кластеров K , которые должны быть найдены. Эти кластеры представляют собой области сегментации.

- **Инициализация центров кластеров.** Изначально выбираются случайные центры этих кластеров.
- **Присваивание пикселей кластерам.** Каждый пиксель изображения присваивается ближайшему кластеру, основываясь на мере расстояния (обычно используется евклидово расстояние), создавая сегментированное изображение.
- **Пересчет центров кластеров.** После того как все пиксели будут присвоены кластерам, вычисляются новые центры для каждого кластера. Новый центр определяется как среднее значение всех пикселей, принадлежащих этому кластеру.
- **Повторение процесса.** Процесс продолжается до тех пор, пока центры кластеров не стабилизируются, то есть пока не прекратятся значительные изменения в их позициях.

Вывод:

Преимущества: метод достаточно эффективен на изображениях с несложной текстурой, где объекты или области имеют схожие цветовые характеристики или текстуры.

Недостатки: случайная инициализация начальных центров кластеров и необходимость знания их количества перед применением делают метод не универсальным. Для изображений с большим количеством сложных объектов или текстур К-средних может не быть достаточно мощным методом.

2.1.2. Метод выделения границ

Методы выделения границ (или контуров) изображений предназначены для нахождения и выделения переходов между различными областями изображения, где происходит резкое изменение интенсивности пикселей. Это может быть полезно для анализа объектов, их формы и структуры. Одним из самых популярных методов является алгоритм Кэнни, подробно рассмотренный в работе [1].

Основные этапы алгоритма Кэнни:

- **Преобразование цветового пространства** из RGB (Red, Green, Blue) в HSV (Hue, Saturation, Value) или HLS (Hue, Lightness, Saturation), чтобы более эффективно сегментировать объекты на изображении на основе их цвета. Как HSV, так и HLS, в отличие от RGB, лучше разделяют оттенок и яркость, что делает их более подходящим для задач, связанных с цветовой сегментацией.
- **Пороговая бинаризация.** Применяется пороговая бинаризация, чтобы выделить пиксели, которые попадают в заданный диапазон цвета. Это ключевой шаг

в цветовой сегментации, особенно для разметки, которая часто имеет фиксированный цвет.

- **Применение фильтра Гаусса**, который размазывает изображение, устраняя мелкие шумы. Это важно, поскольку шум может привести к ложным границам.
- **Детекция границ**. Применяется детектор границ Кэнни — алгоритм оптимального определения границ, который рассчитывает градиенты интенсивности изображения, а затем с помощью двух порогов удаляет слабые границы, оставляя искомые как пороговые значения.

Вывод:

Преимущества: метод простой и быстрый, хорошо работает на изображениях с простыми текстурами.

Недостатки: часто подвержен шуму в случае недостаточной фильтрации, настройкой параметров для гауссова размытия и пороговых значений не всегда удастся добиться качественного решения. Уязвимость к изменениям освещения, к наличию шумов или объектов, которые схожи с искомым по цвету, делает алгоритм не универсальным.

2.1.3. Итог по классическим методам

Несмотря на простоту и скорость классических методов, их ограниченная точность и зависимость от внешних условий делают их непригодными для сложных задач сегментации дорожной разметки. Кроме того, распознавание разметки только парковочных мест невозможно из-за ограничений этих алгоритмов.

2.2. Нейросетевые методы

Современные подходы к обработке изображений основаны на глубоких нейронных сетях, которые значительно превосходят классические методы по точности, особенно в условиях сложных сцен.

2.2.1. Основные типы нейронных сетей и их задачи

- **Полносвязные нейронные сети (Fully Connected Networks, FCNN, см. обзор в работе [5]).** В таких сетях каждый нейрон в одном слое соединён со всеми нейронами в следующем. Эти сети часто применяются для задач классификации данных, где входные данные имеют фиксированную размерность. Их обычно используют для задач обработки табличных данных, прогнозирования или анализа данных, где пространственные связи не играют существенной роли. Полносвязные сети плохо подходят для обработки изображений, поскольку они

не учитывают пространственные зависимости между пикселями, что критично для понимания структуры изображения.

- **Рекуррентные нейронные сети (Recurrent Neural Networks, RNN, см. обзор в работе [6]).** RNN обрабатывают последовательные данные (например, текст или временные ряды), передавая информацию от одного временного шага к другому. Они эффективны для задач, связанных с временной последовательностью, таких как распознавание речи, перевод текста и обработка временных рядов. RNN редко используются для анализа изображений, поскольку не могут эффективно учитывать пространственные взаимосвязи, которые важны для распознавания объектов и их особенностей на изображении.
- **Свёрточные нейронные сети (Convolutional Neural Networks, CNN, см. обзор в работе [9]).** CNN используют свёртки (фильтры), чтобы выделять иерархические признаки на изображении — от простых, таких как контуры, до сложных, таких как формы и текстуры. Они являются стандартом для задач обработки изображений и подходят для классификации, детекции объектов и сегментации изображений. CNN хорошо работают с пространственными зависимостями, обладают высокой точностью в распознавании объектов и их особенностей и поддерживают многослойные структуры для последовательного извлечения признаков.
- **Сети генеративного типа (Generative Networks, см. обзор в работе [4]).** Генеративные сети, такие как вариационные автоэнкодеры и генеративно-состязательные сети (GANs), созданы для генерации новых данных, основанных на распределении существующих данных. Эти сети применяются в задачах генерации изображений, видеоданных, воссоздания недостающих частей изображений или улучшения их разрешения. Генеративные сети используются не для анализа, а скорее для создания и модификации данных, поэтому их эффективность для задач классификации и сегментации дорожной разметки ограничена.

Для задач обработки изображений предпочтение отдается CNN. Основные компоненты CNN:

- **Сверточные слои:** основные слои, которые применяют фильтры к изображению, проходя по нему «окном», на каждом участке изображения операция свертки поэлементно умножает элементы фильтра на элементы изображения, а затем результаты суммируются. Это создаёт значение для активационной карты (feature map), показывающее, насколько данный участок изображения соответствует фильтру. Большое количество сверток позволяет модели учить более сложные и абстрактные признаки, что может улучшить точность, особенно на сложных задачах.

- **Фильтры (ядра):** обычно имеют форму квадратных матриц (например, 3×3 , 5×5). Это наборы весов, которые обучаются для выявления специфических признаков на изображении. Один фильтр может «распознавать» горизонтальные линии, другой — углы и т. д. Все ядра сверток являются обучаемыми параметрами. Сеть автоматически подбирает нужные фильтры для выделения полезных паттернов, таких как края, текстуры и формы. По мере обучения свертки «подстраиваются» друг под друга для выявления сложных зависимостей и признаков.
- **Шаг (Stride):** определяет, на сколько пикселей смещается фильтр при каждом новом «сканировании» изображения. Например, при шаге 1 фильтр двигается на один пиксель, при шаге 2 — на два. Увеличение шага снижает разрешение активационной карты, но уменьшает количество вычислений.
- **Паддинг (Padding):** можно заметить, что операция свертки уменьшает изображение. Также пиксели, которые находятся на границе изображения, участвуют в меньшем количестве сверток, чем внутренние. В связи с этим в сверточных слоях используется padding. Padding — это добавление пикселей по краям изображения, чтобы контролировать размер выходной карты активации и учесть информацию с границ изображения. Обычно эти добавленные пиксели заполняются нулями (zero-padding), но могут использоваться и другие значения, в зависимости от задачи.
- **Слои пулинга:** эти слои уменьшают размер карты признаков (чаще всего с помощью max-pooling), что снижает вычислительные затраты, добавляет инвариантность к сдвигам и помогает избежать переобучения.
- **3D-свертка:** цветные изображения представляют собой 3D-матрицы (RGB), где каждый канал (красный, зелёный и синий) обрабатывается соответствующей плоскостью фильтра. 3D-фильтры применяются ко всем трём каналам изображения, что позволяет учесть цветовую информацию в свёртках.

Архитектуры CNN для обнаружения объектов делятся на одноэтапные и двухэтапные.

Двухэтапные нейросетевые алгоритмы обнаружения объектов на изображении включают в себя следующие два этапа:

Первый этап, который заключается в поиске подозрительных зон на изображении, т. е. зон, возможно содержащих интересующие нас объекты.

Второй этап, на котором CNN классифицирует найденные зоны для определения объектов.

Одноэтапные алгоритмы сразу обнаруживают объекты, без отдельного поиска зон. Их преимущество — простота и скорость, однако точность и гибкость ниже, чем у двухэтапных алгоритмов.

Свёрточные нейронные сети решают множество задач, связанных с компьютерным зрением. Рассмотрим ключевые задачи, в которых CNN демонстрируют выдающиеся результаты:

- **Классификация изображений:** цель — определить, к какому классу принадлежит изображение. CNN обрабатывает изображение и создаёт карту признаков, по которой классификационный слой (обычно полносвязный) определяет категорию объекта на изображении.
- **Локализация объектов:** цель - определить местоположение объекта на изображении.
- **Детекция объектов:** цель - найти и определить координаты каждого объекта определённого класса на изображении. CNN одновременно выполняет классификацию и предсказывает координаты ограничивающих рамок.
- **Сегментация изображений:** цель - классифицировать каждый пиксель изображения. Сегментация делится на два основных типа: семантическая и инстансная.
 - **Семантическая сегментация:** все объекты одного класса помечаются одинаковым цветом, без различия между различными экземплярами одного и того же объекта.
 - **Инстансная сегментация:** различает объекты одного класса, выделяя их как отдельные экземпляры. Например, на изображении с несколькими автомобилями один и тот же класс (автомобиль) будет разделён на несколько разных экземпляров.

2.2.2. Обзор свёрточных нейросетей

Современные архитектуры нейронных сетей предоставляют разнообразные подходы к решению каких-либо задач, каждый из которых обладает уникальными особенностями. Выбор подходящей модели потребует глубокого понимания архитектур, их компонентов и принципов работы, чтобы понимать, как ее архитектура влияет на точность и скорость работы. Далее рассмотрены факторы, влияющие на производительность, а также объясняется, почему определённые архитектуры могут быть предпочтительнее для конкретных задач.

Архитектуры для задачи сегментации:

Для задач сегментации наиболее популярны следующие архитектуры: U-Net [10], DeepLab v3+ [2] и Mask R-CNN [8].

U-Net — это одноэтапная модель, что значит, что она выполняет все операции за один шаг (выполняет задачу сегментации напрямую, без предварительного этапа обнаружения объектов). U-Net обладает U-образной структурой, сочетающей снижение разрешения (downsampling) и восстановление разрешения (upsampling) соединенные с помощью skip-соединений, которые передают высокоуровневую пространственную информацию (например, текстуры и границы объектов) промежуточных слоев downsampling в соответствующие слои upsampling. Это помогает компенсировать потерю пространственного разрешения, возникающую при понижении масштаба изображения. Это делает U-Net быстрее и точнее на изображениях, где важны границы объектов, но модель может страдать от потери информации при сегментации сложных структур из-за ограничений в обработке сложных деталей. U-Net содержит относительно небольшое количество параметров, что снижает требования к вычислительным ресурсам и ускоряет процесс предсказания сегментации.

DeepLab v3+ — это одноэтапная модель, которая оптимизирована для работы с изображениями, содержащими сложные текстуры и неоднородный фон, поскольку в этой нейросети используются достаточно глубокие сверточные слои. Использование модуля ASPP (Atrous Spatial Pyramid Pooling), который включает несколько параллельных расширенных свёрток, позволяет модели эффективно обрабатывать контекст на разных масштабах и учитывать объекты разного масштаба, не увеличивая параметры.

Mask R-CNN применяет двухэтапный подход с генерацией регионов интереса (Region Proposal Network, RPN) для генерации регионов интереса, то есть сеть выделяет области, где могут находиться объекты, что повышает точность. На первом этапе, RPN генерирует предложения регионов (ограничивающих рамок), где могут находиться объекты. На втором этапе, для каждого региона применяется более детальная сегментация через маски, используя RoI Align, что дает очень точную сегментацию. Однако этот процесс требует больше вычислительных ресурсов и времени для предсказания сегментации.

Архитектуры для задачи детекции:

Для задачи детекции наиболее популярны следующие архитектуры: YOLO (You Only Look Once) [13], SSD (Single Shot MultiBox Detector) [11] и Faster R-CNN [3]. Рассмотрим их особенности.

YOLO (You Only Look Once) — одноэтапная архитектура детекции, ориентированная на высокую скорость. Она делит изображение на сетку, где каждая ячейка предсказывает вероятность объектов и их координаты. Поскольку YOLO выполняет детекцию за один проход, модель демонстрирует высокую производительность, хотя может уступать в точности двухэтапным архитектурам. YOLO использует достаточ-

но глубокую сверточную сеть для выделения признаков. Основная часть архитектуры представлена последовательностью слоев свертки и активаций, что позволяет извлекать признаки разного уровня. В YOLO применяются фильтры разного размера (чаще всего 3x3 и 1x1) для компрессии признаков, при этом используется zero-padding для сохранения размеров, что сохраняет информацию на краях изображения.

SSD (Single Shot MultiBox Detector) также является одноэтапной моделью, она использует многоуровневую архитектуру для выявления объектов разных размеров: высокие слои детектируют крупные объекты, низкие — мелкие. Однако её архитектура менее точно предсказывает границы объектов по сравнению с YOLO. SSD применяет компактные свёртки, что снижает вычислительные затраты, сохраняя при этом основную информацию с помощью паддинга. Отсутствие RPN ускоряет работу, хотя и может ограничить точность на сложных изображениях.

Faster R-CNN — двухэтапная архитектура, что делает её наиболее точной и медленной среди представленных моделей. На первом этапе сеть генерирует регионы интереса. На втором этапе модель уточняет классы объектов и их координаты. В качестве экстрактора признаков (backbone) используются глубокие нейронные сети, такие как ResNet. Это позволяет эффективно извлекать детализированные признаки из изображений. Кроме того, применение регионального пуллинга (RoI Pooling) и zero-padding повышает точность. Faster R-CNN показывает высокую точность на изображениях с насыщенными деталями, однако её скорость работы остаётся относительно низкой.

Результат обзора сверточных нейросетей

На основе рассмотренных архитектур для задач сегментации и детекции можно выделить следующие рекомендации:

- **Если нужен баланс между скоростью и точностью:** оптимальным выбором является использование YOLOv8 и DeepLabV3+. Эти модели обеспечивают хорошее сочетание точности и скорости, что делает их подходящими для реальных приложений, где важен баланс между этими параметрами.
- **Если нужна максимальная скорость:** U-Net и SSD являются наиболее подходящими архитектурами. Они обеспечивают быстрый инференс, но могут уступать в точности по сравнению с более сложными моделями.
- **Если нужна максимальная точность:** Faster R-CNN и Mask R-CNN обеспечивают наивысшую точность сегментации и детекции, но требуют больших вычислительных ресурсов и времени для инференса.

Инструменты для реализации нейросетевого решения

Датасеты:

- Для детекции: поскольку fisheye датасета с парковочными местами найти не удалось, будет адаптироваться датасет «Parking Finder»¹, содержащий снимки изображения вида pinhole. Данный датасет включает изображения с парковочными местами, используемые для обучения моделей детекции. Он охватывает разнообразные сцены и условия, что позволяет нейросети выявлять парковочные места в различных ситуациях.
- Для сегментации: датасет «WoodScape»² [12], содержащий снимки, сделанные fisheye камерой, предоставляет возможность дообучить нейросеть для сегментации дорожной разметки.

Платформы для дообучения:

- **Google Colab:** Бесплатная облачная платформа, предоставляющая доступ к GPU и TPU для ускорения обучения нейросетей. Colab позволяет легко загружать и работать с моделями, обеспечивая интеграцию с Google Drive и возможностью установки необходимых библиотек. Это отличное решение для работы с большими нейросетями и значительными вычислительными нагрузками, особенно для задач, требующих быстрого прототипирования. Платформа Colab предоставляет 4,5 часа по бесплатной подписке и одну видеокарту NVIDIA Tesla T4.
- **Kaggle Kernels:** Платформа для выполнения кода в облаке, также предоставляющая бесплатный доступ к GPU и TPU. Kaggle Kernels позволяет удобно работать с большими датасетами, осуществлять их загрузку и проводить обучение моделей прямо в браузере. Платформа Kaggle предоставляет 30 часов работы в неделю по бесплатной подписке и две видеокарты NVIDIA Tesla T4.

В контексте нашей задачи обе платформы обладают схожими возможностями и отличаются лишь доступным количеством часов. Поэтому для максимального увеличения времени использования вычислительных ресурсов задействуем обе платформы.

Метрика mAP (mean Average Precision) для задачи детекции

Метрика mAP измеряет способность модели точно находить и классифицировать объекты на изображениях. Она рассчитывается как среднее значение точности (Average Precision, AP) по всем категориям объектов. Процесс вычисления включает следующие этапы:

1. Для каждой категории объектов вычисляются *точность* (precision) и *полнота* (recall) при различных порогах уверенности модели.
2. Строится кривая Precision-Recall (PR-кривая), отражающая зависимость точности от полноты.

¹<https://universe.roboflow.com/aiml-the-lebron-project/parking-finder>

²<https://github.com/valeoai/WoodScape>

3. Площадь под PR-кривой для каждой категории определяется как AP:

$$AP = \int_0^1 \text{Precision}(\text{Recall}) d(\text{Recall}). \quad (1)$$

4. Среднее значение AP по всем категориям объектов вычисляется как:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (2)$$

где N — количество классов, AP_i — точность для i -го класса.

Метрика IoU (Intersection over Union) для задачи сегментации

Метрика IoU используется для оценки качества сегментации, измеряя степень совпадения между предсказанной областью и истинной разметкой. Она определяется как отношение площади пересечения предсказанной маски и истинной маски к площади их объединения:

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \quad (3)$$

где A — предсказанная область (маска), B — истинная разметка, $|A \cap B|$ — площадь пересечения, а $|A \cup B|$ — площадь объединения.

Значение IoU варьируется от 0 (нет пересечения) до 1 (полное совпадение). Эта метрика позволяет объективно оценить точность выделения объекта на изображении.

2.3. Результаты обзора

Для решения задачи сегментации дорожной разметки парковочных мест оптимальным является дообучение нейросетевых моделей на специальных наборах данных с fisheye изображениями. Классические подходы использоваться не будут, поскольку их точность и устойчивость уступают современным нейросетевым архитектурам.

3. Описание решения

Поскольку готового fisheye датасета для сегментации дорожной разметки парковочного места найти не удалось, для решения задачи будем следовать следующим пунктам:

- **Обработка pinhole датасета парковочных мест:** преобразуем изображения, применяя fisheye фильтр, и отберём только те, которые были сделаны с камер, установленных на автомобиле, исключив снимки с уличных камер.
- **Детекция парковочных мест на fisheye изображениях:** обучим модель, которая будет определять расположение парковочных мест на fisheye изображении.
- **Семантическая сегментация дорожной разметки на fisheye изображениях:** обучим модель, которая будет получать маску всей дорожной разметки.
- **Объединение результатов моделей, решающих задачу сегментации и детекции:** реализуем алгоритм, который выделит из всей дорожной разметки только ту, которая имеет пересечение с ограничивающей рамкой обнаруженного парковочного места.

В этой работе будет использоваться трансферное обучение (transfer learning), то есть дообучение уже существующих нейросетевых решений. Основные причины выбора такого подхода:

- **Экономия времени и вычислительных ресурсов:** при использовании уже обученной модели с дообучением, мы значительно экономим время и вычислительные ресурсы, так как большая часть весов и признаков уже оптимизированы для базового распознавания объектов.
- **Использование накопленных знаний:** обученные на крупных датасетах модели уже умеют выделять полезные признаки, такие как границы, текстуры и цветовые структуры. Эти признаки часто хорошо переносятся на новые задачи, даже если они не совсем такие же, как в оригинальном датасете.
- **Сокращение требований к количеству данных:** для создания точной модели с нуля нужен большой объем данных. Однако при дообучении модели с предобученными весами требования к объему данных значительно ниже.
- **Устойчивость к переобучению:** дообучение часто снижает риск переобучения, особенно при работе с небольшими датасетами. Модель, обученная с нуля на малом объеме данных, может подстроиться под специфические особенности

датасета, не обобщая их на новые изображения. При дообучении на существующей архитектуре, модель уже обладает знаниями общего характера, что помогает избежать слишком сильного подстраивания под данные.

Реализованное решение

Исходя из имеющихся вычислительных ресурсов и наборов данных, было принято реализовать следующее решение: YOLOv8 для детекции и DeepLabv3+ для сегментации. Эти модели обеспечивают оптимальное соотношение точности и скорости, что делает их подходящими для потенциальной интеграции в автомобиль.

YOLOv8 обучена для распознавания парковочных мест на fisheye изображениях при помощи обработанного датасета «Parking Finder». В сравнении с SSD, YOLOv8 обеспечивает значительно более высокую точность благодаря улучшенной архитектуре, эффективно комбинирующей детекцию и классификацию объектов. При этом YOLOv8 лишь незначительно уступает SSD в скорости работы, что делает её оптимальным выбором для задач, где критически важно соотношение точности и производительности.

DeepLabv3+ была обучена для сегментации всей дорожной разметки на fisheye изображениях при помощи датасета «WoodScape». В отличие от U-Net, эта модель предоставляет более высокую точность сегментации сложных текстур и мелких элементов благодаря использованию архитектуры с каскадными и пространственно-пирамидальными модулями, которые эффективно захватывают как глобальные, так и локальные контексты. Модель обладает средней скоростью работы, что делает её подходящей для использования в реальных условиях, обеспечивая оптимальный баланс между точностью и производительностью.

Объединение результатов: разработан алгоритм, выделяющий из маски дорожной разметки, предсказанной DeepLabV3+, только те области, которые пересекаются с ограничивающими рамками, предсказанными YOLOv8. Это позволяет определить разметку, относящуюся исключительно к парковочным местам.

Демонстрацию работы всей системы можно посмотреть по ссылке: <https://github.com/touge13/recognitionOfRoadMarkingsOfParkingSpaces>.

4. Эксперимент

Для подтверждения эффективности выбранных решений были проведены эксперименты по дообучению и тестированию моделей YOLOv8 и DeepLabv3+.

4.1. Исследовательские вопросы

В рамках экспериментов рассматривались следующие исследовательские вопросы:

- Какую точность демонстрирует YOLOv8 при детекции парковочных мест на изображениях с fisheye искажениями?
- Какую точность сегментации дорожной разметки достигает DeepLabv3+ на fisheye изображениях?
- Насколько предложенные подходы к детекции парковочных мест и сегментации дорожной разметки пригодны для использования в задачах анализа дорожной сцены, учитывая их производительность на независимых датасетах?

4.2. Настройки эксперимента

4.2.1. Датасеты:

- Для задачи детекции использовался модифицированный с использованием fisheye фильтра pinhole датасет «Parking Finder».
- Для задачи сегментации применялся датасет «WoodScape», предоставляющий изображения и аннотации дорожной разметки в условиях fisheye камер.

4.2.2. Модели:

- YOLOv8 использовалась для детекции парковочных мест, обеспечивая быстрое и точное определение ограничивающих рамок.
- DeepLabv3+ применялась для сегментации дорожной разметки, что позволяет учитывать сложные текстуры и мелкие детали изображения.

4.2.3. Настройки обучения DeepLabv3+:

- **Экстрактор признаков (backbone):** ResNet-50 — нейронная сеть глубиной 50 слоёв, которая используется для извлечения признаков из изображения. Этот экстрактор признаков обеспечивает баланс между вычислительной эффективностью и точностью.

- **Размер батча:** 4 — выбран из-за ограничений памяти видеокарты, так как модели сегментации требуют значительных вычислительных ресурсов.
- **Оптимизатор:** Adam — один из самых популярных оптимизаторов, основанный на модификации метода градиентного спуска, учитывающий накопленный градиент для более стабильного и быстрого обучения.
- **Функция потерь:** кросс энтропия. Она измеряет разницу между двумя вероятностными распределениями: истинным распределением меток и предсказанным распределением.
- **Количество эпох:** 20 — достаточно для достижения удовлетворительного результата, учитывая ограниченное время вычислительных ресурсов и размера датасета.
- Для обучения использовались GPU-T4 x2 на платформе Kaggle, поскольку предоставленных платформой Colab часов на использование видеокарты не хватает для дообучения модели.

4.2.4. Настройки обучения YOLOv8:

- **Размер батча:** 32 — большее значение по сравнению с DeepLabv3+ из-за оптимизированного механизма обработки данных и меньших вычислительных требований при обучении детекторов.
- **Оптимизатор:** Auto — автоматический выбор оптимизатора, который подбирает наиболее эффективный метод для текущего датасета и задачи, обеспечивая лучшее сочетание скорости и точности.
- **Функции потерь:** комбинированная, включающая в себя несколько составляющих, каждая из которых отвечает за оптимизацию различных аспектов модели.
 - Для регрессии координат — (IoU/CIoU).
 - Для оценки вероятности наличия объекта — (BCE Loss).
 - Для классификации объектов по классам — (BCE Loss).
- **Количество эпох:** 250 — достаточно для достижения удовлетворительного результата, учитывая ограниченное время вычислительных ресурсов и размер датасета.
- Для обучения использовались GPU-T4 x2 на платформе Kaggle, поскольку предоставленных платформой Colab часов на использование видеокарты не хватает для дообучения модели.

4.3. Результаты эксперимента

- YOLOv8 достигла $mAP50 = 0.75$ на тестовом наборе данных.
- DeepLabv3+ показала $IoU = 0.79$ на тестовом наборе данных.

Несмотря на независимое обучение моделей, их результаты демонстрируют потенциал для последовательного использования в комплексной задаче анализа дорожной сцены. Детекция парковочных мест с помощью YOLOv8 может служить первым этапом, обеспечивая локализацию областей интереса, тогда как сегментация дорожной разметки с помощью DeepLabv3+ позволяет точно анализировать текстуры и структуру дорожной сцены. Теоретически такая связка должна работать корректно, поскольку этапы детекции и сегментации решают взаимодополняющие задачи. Однако качество всей системы измерить нет возможности из-за отсутствия fisheye датасета с размеченной дорожной разметкой парковочного места. Дальнейшая работа предполагает поиск или создание единого датасета, который позволит оценить совмещённое использование выбранных подходов в реальных условиях анализа дорожной сцены.

Заключение

В рамках работы были изучены существующие подходы к решению задач анализа дорожной сцены, что позволило выявить, что использование нейросетевых методов является наиболее оптимальным выбором для данного проекта.

Был реализован модуль распознавания дорожной разметки парковочных мест по fisheye камерам с использованием сверточных нейронных сетей, включая модели YOLOv8 для детекции и DeepLabv3+ для сегментации.

На текущем этапе сравнение с другими подходами пока не выполнено, так как текущие наборы данных и модели, доступные в открытом доступе, не обладают аналогичными характеристиками или аннотациями, подходящими для данного проекта. В дальнейшем планируется провести сравнение, как только будет найден сопоставимый набор данных или опубликованы результаты по схожим задачам.

Демонстрацию работы всей системы можно посмотреть по ссылке: <https://github.com/touge13/recognitionOfRoadMarkingsOfParkingSpaces>.

Потенциальные доработки

В перспективе проект может быть доработан следующими способами:

- **Приобретение fisheye датасета:** Приобрести готовый датасет, на котором размечена дорожная разметка парковочных мест на fisheye изображениях, и дообучить на нём DeepLabv3+. Такой подход обладает ключевым преимуществом: он позволяет существенно сэкономить вычислительные ресурсы, необходимые для детекции парковочных мест, ведь задача сегментации будет решаться напрямую. Это особенно важно в условиях ограниченных вычислительных ресурсов, обеспечивая оптимальный баланс между точностью и скоростью.
- **Расширение вычислительных ресурсов:** Инвестировать в дополнительные вычислительные мощности, чтобы создать собственный высококачественный датасет с использованием Mask R-CNN и Faster R-CNN. Несмотря на низкую скорость работы, эти модели обеспечивают максимальную точность. На этом датасете, обладая большим количеством вычислительных мощностей, можно обучить U-Net и в итоге добиться приемлемой точности и максимальной скорости работы.

Список литературы

- [1] Canny John. A Computational Approach to Edge Detection // [IEEE Transactions on Pattern Analysis and Machine Intelligence](#). — 1986. — Vol. 8, no. 6. — P. 679–698.
- [2] Chen Liang-Chieh, Zhu Yukun, Papandreou George, Schroff Florian, and Adam Hartwig. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation // arXiv preprint arXiv:1802.02611. — 2018. — Access mode: <https://arxiv.org/abs/1802.02611> (online; accessed: 15 декабря 2024 г.).
- [3] Ren Shaoqing, He Kaiming, Girshick Ross, and Sun Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // arXiv preprint arXiv:1506.01497. — 2015. — Access mode: <https://arxiv.org/abs/1506.01497> (online; accessed: 15 декабря 2024 г.).
- [4] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, and Bengio Yoshua. Generative Adversarial Networks // arXiv preprint arXiv:1406.2661. — 2014. — Access mode: <https://arxiv.org/abs/1406.2661> (online; accessed: 15 декабря 2024 г.).
- [5] Goodfellow Ian, Bengio Yoshua, and Courville Aaron. Deep Learning. — MIT Press, 2016. — Access mode: <https://www.deeplearningbook.org/> (online; accessed: 15 декабря 2024 г.).
- [6] Lipton Zachary C., Berkowitz John, and Elkan Charles. A Critical Review of Recurrent Neural Networks for Sequence Learning // arXiv preprint arXiv:1506.00019. — 2015. — Access mode: <https://arxiv.org/abs/1506.00019> (online; accessed: 15 декабря 2024 г.).
- [7] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. — 1967. — Vol. 1. — P. 281–297. — Access mode: <https://projecteuclid.org/euclid.bsmsp/1200512992> (online; accessed: 15 декабря 2024 г.).
- [8] He Kaiming, Gkioxari Georgia, Dollár Piotr, and Girshick Ross. Mask R-CNN // [IEEE Transactions on Pattern Analysis and Machine Intelligence](#). — 2020. — Vol. 42, no. 2. — P. 386–397. — Access mode: <https://arxiv.org/abs/1703.06870> (online; accessed: 15 декабря 2024 г.).
- [9] Gu Jiuxi, Wang Ziang, Kuen Jason, et al. Recent Advances in Convolutional Neural Networks // [Pattern Recognition](#). — 2018. — Vol. 77. — P. 354–377. — Access mode: <https://arxiv.org/abs/1512.07108> (online; accessed: 15 декабря 2024 г.).

- [10] Ronneberger Olaf, Fischer Philipp, and Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation // arXiv preprint arXiv:1505.04597. — 2015. — Access mode: <https://arxiv.org/abs/1505.04597> (online; accessed: 15 декабря 2024 г.).
- [11] Liu Wei, Anguelov Dragomir, Erhan Dumitru, Szegedy Christian, Reed Scott, Fu Cheng-Yang, and Berg Alexander C. SSD: Single Shot MultiBox Detector // arXiv preprint arXiv:1512.02325. — 2016. — Access mode: <https://arxiv.org/abs/1512.02325> (online; accessed: 15 декабря 2024 г.).
- [12] Yogamani Senthil, Kruber Michael, Simon Marius, , et al. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving // Proceedings of the IEEE International Conference on Computer Vision (ICCV). — 2019. — Access mode: <https://arxiv.org/abs/1905.01489> (online; accessed: 15 декабря 2024 г.).
- [13] Jocher Glenn, Chaurasia Akhilesh, Qiu Jing, and Fang Changyong. YOLOv8: State-of-the-Art Object Detection and Segmentation // Ultralytics Blog. — 2023. — Access mode: <https://ultralytics.com/yolov8> (online; accessed: 15 декабря 2024 г.).