МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

Институт компьютерных наук НИТУ МИСИС

Кафедра инженерной кибернетики

СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ» НА ТЕМУ «ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ПРОМЫШЛЕННЫХ, КОММЕРЧЕСКИХ, МЕДИЦИНСКИХ И ФИНАНСОВЫХ ПРИЛОЖЕНИЯХ»

Москва, 2025

УДК 004.8
ББК 32.813.5

Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях, 2025: Сборник статей научно-технического семинара. Вып. 3 / Под ред. А.Р. Ефимова— М.: НИТУ «МИСИС», 2025.— 142 с.: табл., ил., цв. ил.

Настоящий сборник содержит материалы научно-технического семинара «Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях», организатором которой является кафедра Инженерной кибернетики Института компьютерных наук НИТУ «МИСИС». На семинаре были представлены доклады по применению искусственного интеллекта в различных задачах народного хозяйства: промышленных, коммерческих, медицинских и финансовых приложениях.

Семинар проходил 26-27 декабря 2024 г. в режиме онлайн.
Редакционная коллегия: Ефимов А.Р., Бакулев К.С., Садеков Р.Н., Мишуров С.С.
Редактор: Садеков Р.Н.
Компьютерная верстка: Садеков Р.Н.

Рецензенты: Садеков Р.Н. д.т.н., доцент, профессор кафедры инженерной кибернетики НИТУ «МИСИС», Тарханов И.А. к.т.н., доцент кафедры инженерной кибернетики НИТУ «МИСИС», Курочкин И.И. к.т.н, доцент кафедры инженерной кибернетики НИТУ «МИСИС».

Содержание

ГРНТИ 20.53.21, 20.53.23, 20.53.25
ASJC 1707, 1712, 1706
OECD 01.02.EP

Распознавание самокатов в реальном времени с помощью YOLO: сравнительный анализ YOLOv10, YOLOv11

Л.Е. Алексеев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2413819@edu.misis.ru

***Аннотация*— в последние годы электросамокаты приобретают всё большую популярность в городских условиях, что повышает требования к системам обеспечения их безопасности. Одной из ключевых задач является фиксирование нарушений правил использования индивидуальных транспортных средств, для этого необходимо детектирование самокатов. В данной работе проводится сравнительный анализ двух современных моделей детектирования объектов — YOLOv10 и YOLOv11 — в контексте распознавания электросамокатов. Оцениваются их архитектурные особенности, точность обнаружения (mAP), скорость инференса и вычислительная эффективность. Экспериментальные результаты показывают преимущества каждой модели, что позволяет определить наиболее подходящую архитектуру для применения в системах безопасности электросамокатов.**

Ключевые слова — компьютерное зрение, распознавание электросамокатов, YOLOv10, YOLOv11, mAP, реальное время, индивидуальные транспортные средства

С увеличением числа электросамокатов в городах возрастает потребность в системах, способных фиксировать нарушения их использования. Это включает контроль за соблюдением правил, таких как: ограничение скорости, запрет на совместную езду вдвоём и использование в зонах с ограниченным движением.

Электросамокаты часто создают сложности для других участников дорожного движения из-за высокой манёвренности и разнообразия скоростей. Поэтому важно разрабатывать системы, которые могут в реальном времени обнаруживать и отслеживать такие средства передвижения.

Современные методы, основанные на алгоритмах глубокого обучения, особенно модели детектирования объектов, такие как YOLO (You Only Look Once), показывают хорошие результаты в подобных задачах. В этой работе рассматриваются две последние версии моделей — YOLOv10 и YOLOv11. Проводится их сравнение по точности, скорости работы и эффективности для детектирования электросамокатов с целью последующей фиксации нарушений.

I. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались один личный и один открытый набор данных

A. Публичный набор данных (ScooterDet)

Набор данных ScooterDet содержит 2013 изображений с 11 классами дорожных объектов: "человек", "автомобиль", "грузовик", "автобус", "светофор", "пожарный гидрант", "знак стоп", "скамейка" и "самокат". Всего в наборе 11 011 аннотаций ограничивающих рамок. Данные собраны с помощью очков Tobii Pro Glasses 2, установленных на самокате Segway NineBot, при движении от кампуса Университета Виргинии до городской зоны Шарлотсвилл, штата Виргинии, США. Изображения извлечены из видеозаписей, снятых этими очками, и очищены от низкокачественных кадров и изображений без релевантных объектов.

Изначально набор данных содержал аннотации в формате JSON, созданные с помощью инструмента разметки LabelMe. Однако для обеспечения совместимости с фреймворком обучения YOLO потребовалась конвертация аннотаций в стандартный формат YOLO. Этот процесс включал преобразование координат ограничивающих рамок и классов объектов в числовые значения, соответствующие требованиям формата YOLO.

Представили датасет в стандартной разметке YOLO:

- Класс объекта (номер класса).
- Координаты центра ограничивающей рамки (x, y) в нормализованных значениях.
- Ширина и высота рамки относительно размеров изображения.

После преобразования данные были случайным образом разделены на три подмножества:

- Обучающая выборка – 60% (1207 изображений)
- Валидационная выборка – 20% (402 изображения)
- Тестовая выборка – 20% (404 изображения)

На следующем этапе данные были дополнительно обработаны (аугментация), чтобы повысить их разнообразие и устойчивость к реальным условиям, включая изменения освещения, ракурса и фона.

Модели YOLO проходили обучение с использованием заранее натренированных весов (transfer learning), полученных на наборе данных COCO. Изображения были приведены к разрешению 640 × 640 пикселей, соответствующему входным требованиям YOLO.

Каждое изображение сопровождается текстовым файлом аннотаций, где указаны:

- Класс объекта (номер класса).
- Координаты центра ограничивающей рамки (x, y) в нормализованных значениях.
- Ширина и высота рамки относительно размеров изображения.

Рис. 1. Примеры изображений

B. Собственная выборка

Для исследования была создана собственная выборка данных, собранная с бортовых камер автомобилей Tesla, передвигавшихся по городу Москва. Исходные данные включали около 128 ГБ видеоматериалов, снятых в реальных городских условиях. Видеоматериалы содержали эпизоды с электрическими самокатами в различных ситуациях:

- Стоящие отдельно – как в специально отведённых местах, так и в неположенных местах.
- Находящиеся в использовании – в движении под управлением человека.

На первом этапе данные были извлечены из видео в виде отдельных кадров с разрешением 1920 × 1080 пикселей. Для повышения качества были удалены кадры с плохим освещением, размытием и отсутствием объектов интереса.

На следующем этапе предобработки ~~мы~~ снизили качество входящих изображений, уменьшив их разрешение до 640 × 640 пикселей, чтобы привести их в соответствие с характеристиками популярных публичных датасетов, таких как COCO и ScooterDet. Этот шаг был необходим для обеспечения сопоставимости наших данных с открытыми наборами и упрощения последующего обучения моделей детекции объектов.

Для разметки данных был использован инструмент **CVAT** (Computer Vision Annotation Tool), который позволяет создавать высокоточные аннотации с использованием удобного интерфейса. Все объекты, включая электрические самокаты (в стоячем положении и в движении), транспортные средства, а также пешеходов, были размечены вручную в формате **YOLO**. Каждое изображение получило аннотации с координатами ограничивающих рамок, классами объектов и размерами рамок в нормализованных значениях.

Рис. 2. Примеры изображений отдельно стоящего электросамоката

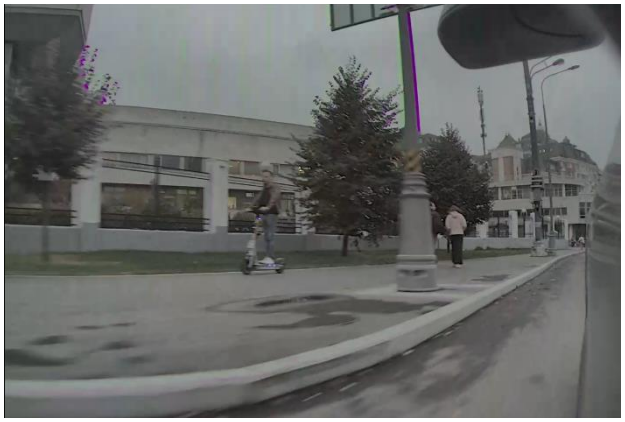


Рис. 3. Примеры изображений электросамоката в использовании

II. Архитектурные особенности нейросетей YOLO

A. YOLOv10

YOLOv10 представляет собой усовершенствованную версию алгоритмов серии YOLO, ориентированную на повышение точности и скорости обнаружения объектов. Эта модель адаптирована для работы в условиях, требующих высокой производительности, например, при детекции электрических самокатов в городской среде и фиксировании нарушений.

- Улучшенная структура сети: использует улучшенный Backbone, основанный на CSPNet (Cross Stage Partial Network). Это повышает эффективность извлечения признаков и снижает избыточность вычислений.
- Spatial Pyramid Pooling-Fast (SPPF): добавлен блок для обработки мульти-масштабных признаков, что улучшает способность обнаружения объектов разного размера, включая мелкие и частично перекрытые объекты, такие как самокаты.
- Consistent Dual Assignments: новый подход к обучению, исключающий необходимость использования Non-Maximum Suppression (NMS). Это упрощает процесс обнаружения и снижает время инференса.

- Эффективный дизайн головы сети: оптимизированная архитектура головы сети уменьшает число параметров и вычислительных операций, сохраняя высокую точность.

B. *YOLOv11*

YOLOv11 — это дальнейшее развитие серии YOLO с акцентом на повышение точности и возможностей обнаружения объектов в сложных условиях.

Основные улучшения YOLOv11:

- C3k2 блоки: внедрение блоков Cross Stage Partial с уменьшенным размером ядра. Это повышает скорость обработки и снижает количество параметров, сохраняя при этом высокую точность.
- Parallel Spatial Attention (C2PSA): новый механизм пространственного внимания, улучшающий фокусировку модели на ключевых областях изображения. Это особенно полезно для поиска мелких или частично скрытых объектов.
- Оптимизированная голова сети: использует усовершенствованные блоки Conv-BatchNorm-SiLU (CBS), что повышает стабильность обучения и качество предсказаний.
- Расширенная функциональность: YOLOv11 поддерживает дополнительные задачи компьютерного зрения, включая:
    - Сегментацию экземпляров.
    - Оценку поз.
    - Обнаружение ориентированных объектов (OBB).

III. СРАВНЕНИЕ

A. *Результаты*

Для детектирования электросамокатов в реальном времени был проведён сравнительный анализ моделей YOLOv10 и YOLOv11 на основе следующих критериев:

1. Точность обнаружения (mAP):
    - YOLOv10s: mAP@0.5 = 0.814, mAP@0.5-0.95 = 0.388
    - YOLOv11s: mAP@0.5 = 0.839, mAP@0.5-0.95 = 0.406
2. Скорость инференса:
    - YOLOv10s: 1.7 мс на изображение
    - YOLOv11s: 1.4 мс на изображение
3. Сложность модели (параметры и GFLOPs):
    - YOLOv10s: 8,044,248 параметров, 24.5 GFLOPs
    - YOLOv11s: 9,417,444 параметров, 21.3 GFLOPs

Рис. 4. Примеры детекции на тестовой выборке

B. *Точность*

YOLOv11 демонстрирует улучшение точности обнаружения по сравнению с YOLOv10, особенно в категории электросамокатов (сравнивания точность между классами). Увеличение точности достигается за счёт более продвинутых архитектурных блоков и механизмов внимания, позволяющих модели лучше фокусироваться на важных деталях изображения.

C. *Скорость*

Несмотря на увеличение количества параметров, YOLOv11 обеспечивает более быструю обработку изображений. Оптимизация архитектуры головы сети и использование эффективных блоков позволяет снизить время инференса без потери точности.

D. *Вычислительная эффективность*

YOLOv11, хотя и имеет больше параметров, оптимизирована по GFLOPs, что позволяет ей быть более эффективной в вычислительном плане.

IV. ЗАКЛЮЧЕНИЕ

В данной работе были подготовлены и использованы два набора данных — публичный ScooterDet и собственная выборка, собранная в Москве с камер автомобилей Tesla. Оба датасета были предобработаны, аннотированы и приведены к формату YOLO для обучения моделей детектирования. На полученных тестовых выборках были обучены последние модели YOLO.

Далее, мы провели сравнительный анализ эффективности по разным параметрам обученных моделей в контексте обнаружения электросамокатов в реальном времени. Экспериментальные результаты показывают, что YOLOv11 превосходит YOLOv10 по показателям точности и скорости инференса, несмотря на увеличение числа параметров. Это достигается благодаря улучшенной архитектуре, использованию блоков внимания и оптимизации вычислительных операций.

ЛИТЕРАТУРА

[1] Neha Dwivedi (2024) "YOLOv11: The Next Leap in Real-Time Object Detection", Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2024/10/yolov11/ (Accessed: November 15, 2024).

[2] Arya, S. (2024) "YOLOv10: Revolutionizing Real-Time Object Detection", Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2024/07/yolov10-for-realtime-object-detection/ (Accessed: December 5, 2024).

[3] Кирвяков, В. О. Исследование возможности детектирования трещин и дорожных заплаток на асфальте / В. О. Кирвяков // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 65-70. – EDN OBGTBO.

[4] Eipifanov, V.A. (2019) Application of Artificial Intelligence and Machine Vision Methods in the Task of Detecting Moving Objects for Determining Dynamic Characteristics of Traffic Flows, Eurasian Union of Scientists. Available at: https://cyberleninka.ru/article/n/primenenie-metodov-iskusstvennogo-intellekta-i-mashinnogo-zreniya-v-zadache-detektirovaniya-obektov-dvizheniya-dlya-dalneyshego (Accessed: October 22, 2024).

[5] Kapsky, D.V., Levanovich, D.V., Ivanov, V.P., and Golovnich, A.K. (2022) Analysis of Traffic Flow Parameter Detection, Bulletin of Polotsk State University. Available at: https://cyberleninka.ru/article/n/analiz-detektirovaniya-parametrov-dorozhnogo-dvizheniya (Accessed: November 8, 2024).

[6] Kaluzhny, Y.N. (2019) Modern Problems of Legislative Regulation for the Use of Specific Types of Electric Transport, NB: Administrative Law and Administration Practice. Available at: https://cyberleninka.ru/article/n/sovremennye-problemy-zakonodatelnogo-regulirovaniya-ispolzovaniya-otdelnyh-vidov-elektrotransporta (Accessed: December 18, 2024).

[7] Карякин, А. В. Исследование возможности классификации дорожных знаков / А. В. Карякин // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 59-61. – EDN FEGRPY.

[8] Ultralytics (2024) "YOLOv10 - Ultralytics YOLO Docs". Available at: https://docs.ultralytics.com/models/yolov10/ (Accessed: November 29, 2024).

[9] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., and Ding, G. (2024) "YOLOv10: Real-Time End-to-End Object Detection", arXiv. Available at: https://arxiv.org/abs/2405.14458 (Accessed: October 30, 2024).

[10] Viso.ai (2024) "YOLO Explained: From v1 to v11". Available at: https://viso.ai/computer-vision/yolo-explained/ (Accessed: December 12, 2024).

[11] Roboflow (2024) "YOLO11 vs. YOLOv10: Compared and Contrasted". Available at: https://roboflow.com/compare/yolo11-vs-yolov10 (Accessed: November 22, 2024).

[12] Paperspace (2024) "YOLOv10: Advanced Real-Time End-to-End Object Detection". Available at: https://blog.paperspace.com/yolov10-advanced-real-time-end-to-end-object-detection/ (Accessed: October 25, 2024).

[13] Butenko, V.V. (2015) Object Search in Images Using Adaptive Enhancement Algorithm, Young Scientist. Available at: https://moluch.ru/archive/84/15604/ (Accessed: November 3, 2024).

[14] Volkov, A.K. (2018) Master's Thesis, Moscow Aviation Institute. Available at: https://www.mai.moscow/download/attachments/50888733/%D0%94%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F_%D0%92%D0%BE%D0%BB%D0%BA%D0%BE%D0%B2%D0%90%D0%9A_%D0%9C%D0%90%D0%98_%D0%9C80_203%D0%9C_18.pdf (Accessed: October 28, 2024).

[15] Strategy Journal (2024) Regulation and Safe Development of New Urban Transport - Scooters. Available at: https://strategyjournal.ru/gosudarstvo/regulirovanie-i-bezopasnoe-razvitie-novogo-gorodskogo-transporta-samokatov/ (Accessed: November 10, 2024).

[16] Wang, C.-Y., and Liao, H.-Y. M. (2024) "YOLOv1 to YOLOv10: The Fastest and Most Accurate Real-Time Object Detection Systems", arXiv. Available at: https://arxiv.org/abs/2408.09332 (Accessed: November 5, 2024).

ГРНТИ 28.23.37
ASJC 1702
OECD 01.02.EP

Сравнение и анализ моделей FCN и DeepLabV3 в задаче семантической сегментации объектов городской улицы

К. В. Андронов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2009115@edu.misis.ru

Аннотация — в данной статье представлен анализ и сравнение двух нейросетевых архитектур Fully Convolutional Network (FCN) на основе ResNet-50 и DeepLabV3 с той же базовой архитектурой — для задачи семантической сегментации объектов на уличных изображениях, используя датасет CamVid [1]. Исследование включает детальное описание архитектур моделей, подход к обработке данных, а также методологию экспериментов, включая использование комбинированной функции потерь Jaccard и Dice, ранней остановки и оценки ключевых метрик (IoU, Dice, Pixel Accuracy) [2]. Результаты экспериментов продемонстрировали, что FCN превосходит DeepLabV3 на данном ограниченном по объему датасете, обеспечивая лучшие метрики IoU (0.324 против 0.200) и Dice (19.006 против 17.876). Графические визуализации масок подтверждают численные выводы, демонстрируя более качественную сегментацию для FCN. Основным выводом работы является то, что для узких задач с малым объемом данных предпочтительны менее сложные архитектуры, такие как FCN, поскольку они менее зависимы от объема тренировочного набора и более устойчивы к ограничениям ресурсов.

I. Введение

Семантическая сегментация является ключевой задачей в области компьютерного зрения, так как позволяет определить принадлежность каждого пикселя изображения к определенному классу. Это особенно важно для автономного вождения, где требуется точная идентификация объектов на дорожной сцене, таких как автомобили, пешеходы, светофоры и дорожные разметки. Прогресс в глубоком обучении привел к появлению мощных архитектур, способных решать подобные задачи с высокой точностью, а также более сложных систем, способных к самостоятельной навигации, исходя из обнаруженных объектов в поле зрения [3, 19, 20, 21].

В настоящем исследовании рассматриваются и сравниваются две популярные архитектуры для семантической сегментации: FCN и DeepLabV3. Первая модель основана на концепции полностью сверточных сетей (Fully Convolutional Networks, FCN), где используются сверточные операции для трансформации изображений в пространстве признаков. Вторая модель, DeepLabV3, применяет усовершенствованный подход с использованием атрибуции пространственного пирамидального пула (Atrous Spatial Pyramid Pooling, ASPP) для захвата контекста на различных масштабах [4][5].

Датасет CamVid, использованный для данного исследования, содержит аннотированные видеокадры дорожных сцен, предоставляющие полезный контекст для обучения и тестирования моделей. Он включает 32 класса объектов, что делает его универсальным инструментом для анализа городских сценариев [6]. Благодаря высокому качеству данных и разнообразию сцен, CamVid стал стандартом для оценки производительности моделей семантической сегментации.

Цель исследования заключается в анализе производительности архитектур FCNResNet50 и DeepLabResNet50 на основе метрик точности, таких как mIoU (mean Intersection over Union), а также выявлении их преимуществ и недостатков в условиях реальных дорожных сцен.

II. Обзор текущих наработок

Семантическая сегментация за последние годы получила значительное развитие благодаря внедрению глубокого обучения. Наиболее успешные подходы используют сверточные нейронные сети (CNN), которые позволяют извлекать богатые пространственные признаки из изображений и эффективно классифицировать каждый пиксель.

Архитектура FCN, представленная в 2015 году, стала первой моделью, способной производить семантическую сегментацию с помощью полностью сверточных сетей. В данной архитектуре вместо полносвязных слоев используются только свёрточные, что позволяет сохранять пространственную информацию и выполнять сегментацию на пиксельном уровне [7]. Использование ResNet50 в качестве базовой сети улучшает качество извлечения признаков благодаря глубокой архитектуре и наличию остаточных связей, предотвращающих исчезновение градиента [8, 18].

Модель DeepLabV3, представленная в 2017 году, улучшает точность сегментации благодаря введению пространственного пирамидального пула (Atrous Spatial Pyramid Pooling, ASPP). ASPP позволяет учитывать контекст на различных масштабах, что особенно важно для сложных сцен с множеством объектов. Как и в случае FCN, DeepLabV3 эффективно использует ResNet50 в качестве базовой сети для извлечения признаков [9].

Предыдущие исследования продемонстрировали, что обе архитектуры демонстрируют высокие результаты на дорожных сценах. Однако их производительность мо-

жет варьироваться в зависимости от сложности сцены и структуры датасета.

III. Нейросетевые архитектуры

Архитектура Fully Convolutional Networks (FCN) стала важной вехой в развитии семантической сегментации. В основе FCN лежит идея использования полностью сверточных слоев, что позволяет проводить пиксельную классификацию и сохранять пространственные зависимости в изображении. Для экспериментов была выбрана модификация FCN с базовой сетью ResNet50, которая обеспечивает глубокое извлечение признаков за счет остаточных связей [10].

Для адаптации модели к датасету CamVid в финальный слой классификатора был добавлен новый сверточный слой: nn.Conv2d (512, n_class, kernel_size=1), где n_class соответствует числу классов в датасете. Этот слой преобразует выходные признаки модели в вероятность принадлежности пикселя к одному из 32 классов. Все остальные слои архитектуры были оставлены неизменными. Архитектура FCN на основе ResNet-50[9]:

Рис. 1. Архитектура FCN на основе ResNet-50

Архитектура DeepLabV3 расширяет возможности FCN благодаря внедрению атрибуции пространственного пирамидального пула (Atrous Spatial Pyramid Pooling, ASPP), что позволяет модели эффективно учитывать контекст на разных масштабах. Это особенно важно для сложных сцен с множеством объектов. Как и в случае FCN, базовой сетью в DeepLabV3 служит ResNet50 [11]. Адаптация DeepLabV3 для работы с датасетом CamVid включала добавление нового слоя классификатора: nn.Conv2d (256, n_class, kernel_size=1). Кроме того, из-за ограниченного размера датасета все слои BatchNorm были заменены на GroupNorm [12]. Это решение было обусловлено тем, что BatchNorm требует батчи больше единицы для корректной работы, что не всегда возможно при работе с небольшими наборами данных. Архитектура DeepLabV3 на основе ResNet-50:

Рис. 2. Архитектура DeepLabV3 на основе ResNet-50

Обе модели используют ResNet50 как основу, что обеспечивает высокий уровень извлечения признаков. Однако подходы к обработке пространственного контекста отличаются. FCN выполняет интерполяцию пространственных признаков для восстановления оригинального размера изображения. DeepLabV3 задействует ASPP, что улучшает сегментацию объектов мелкого масштаба и обеспечивает устойчивость к изменениям в размере объектов. Таким образом, обе архитектуры демонстрируют потенциал для семантической сегментации, но различаются в подходах к обработке контекста и масштабируемости.

IV. Описание датасета

Датасет CamVid представляет собой один из наиболее часто используемых наборов данных для задачи семантической сегментации, особенно в области автономного вождения и анализа уличных сцен. Он был собран в рамках Cambridge-driving Labeled Video Database и состоит из видеокадров, снятых на дорогах, с аннотированными объектами, принадлежащими к 32 различным классам. Эти классы включают различные типы объектов, такие как автомобили, пешеходы, здания, дорожные разметки и другие элементы городской инфраструктуры [13]. Датасет включает в себя изображения, сопровождающиеся масками сегментации, где каждому пикселю присвоен один из 32 классов. Важно отметить, что каждый класс в датасете имеет свой уникальный цветовой код в формате RGB, что позволяет легко ассоциировать каждый объект с его соответствующей категорией. Пример структуры классов приведен в файле class_dict.csv, который содержит информацию о каждом классе и его цвете [2]. Размеры датасета: 367 пар обучающих изображений и соответствующих масок. 101 пара валидационных изображений и масок. 233 пары тестовых изображений и масок.

Рис. 3. Пример изображения из датасета

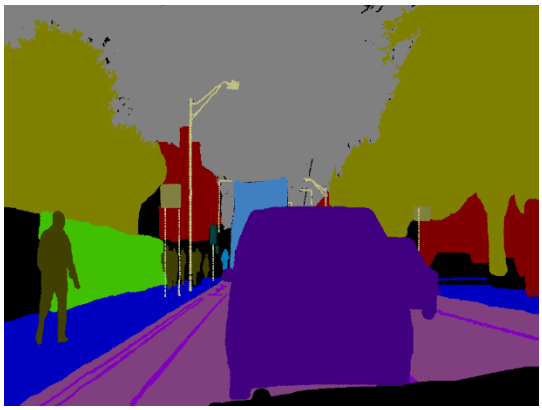


Рис. 4. Пример таргет маски из датасета.

Также к датасету приложен файл маппинга цветов и меток класса, что в дальнейшем эксперименте позволило предсказывать сразу метки класса, предварительно конвертировав цвета пикселей на изображениях в номерные метки классов.

V. Методология эксперимента

Для входных данных были использованы изображения из датасета CamVid, приведенные к разрешению 512x512 пикселей. Предварительная обработка включала следующие трансформации: Resize, Normalize, ToTensorV2. Маски изображений, содержащие цветовые коды классов, были преобразованы в метки классов [14]. Для этого пиксели с определенным RGB-значением из масок сопоставлялись с индексами соответствующего класса на основе заданного цветового маппинга.

Для обучения использовалась совмещенная функция потерь Jaccard и Dice, которая одновременно учитывает пересечение объектов и сбалансированность метрик. Оптимизатор Adam использовался без адаптивного изменения learning rate [15].

Обучение каждой модели сопровождалось использованием механизма early stopping. Обучение прекращалось, если loss на валидационном наборе не уменьшался в течение трех последовательных эпох. FCN завершила обучение за 32 эпохи. DeepLabV3 завершила обучение за 28 эпох. На каждой эпохе оценивались следующие метрики:

- Loss на валидационном наборе.
- IoU (Intersection over Union), измеряющий пересечение между предсказанными и истинными областями.
- Dice — метрика, отражающая схожесть предсказанной и реальной сегментации.
- Pixel Accuracy — доля правильно классифицированных пикселей.

Дополнительно проводилась тестовая валидация после завершения обучения для сравнения качества моделей на тестовом наборе данных.

VI. Результаты эксперимента

На предоставленных изображениях 5 и 6 визуализированы результаты сегментации моделей FCN и DeepLabV3. Оригинальное изображение, разметка и предсказанная маска представлены для обеих моделей.

Рис. 5. Результаты FCN

Рис. 6. Результаты DeepLabV3

Для FCN наблюдаются следующие особенности: модель достаточно точно выделяет ключевые области, такие как здания, дороги и транспортные средства. Однако границы объектов зачастую размыты, а мелкие элементы (например, светофоры или столбы) плохо сегментируются. Более мелкие объекты поглощаются крупными фоновыми объектами, например столбы на фоне стен здания. Для DeepLabV3: качество сегментации заметно ниже, особенно для мелких объектов и деталей. Сильно выражены артефакты на предсказанных масках, где объекты либо сливаются с фоном, либо их форма и контуры искажены. На обоих примерах видно, что DeepLabV3 хуже справляется с задачей точного выделения классов. Эта модель способна выделять общие контуры крупных классов таких, как небо, дома, дорога, но более мелкие объекты не способна выделять. Например, пешеход не был обнаружен совсем.

Что подтверждает предположение о том, что данная архитектура более требовательна к количеству данных для обучения. Результаты метрик на тестовом наборе демонстрируют превосходство FCN над DeepLabV3[16]:

Таблица 1 – метрики эксперимента

	Метрики		
Модель	IoU	Dice	Pixel Accuracy
FCN	0.324	19.006	0.841
DeepLabV3	0.200	17.876	0.758

IoU (Intersection over Union) показывает лучшее покрытие предсказанных областей для FCN. Dice метрика, измеряющая сходство предсказанных и истинных масок, также выше для FCN. Pixel Accuracy подтверждает, что FCN точнее классифицирует отдельные пиксели.[17]

Результаты подтверждают, что FCN более эффективно справляется с задачей семантической сегментации на небольших и специфических датасетах, таких как CamVid. Причины могут быть следующими:

- FCN менее требовательна к количеству данных, поскольку её архитектура использует меньше параметров по сравнению с DeepLabV3.
- DeepLabV3 имеет более сложные механизмы, такие как ASPP (Atrous Spatial Pyramid Pooling), которые требуют больших объемов данных для стабильного обучения.
- Замена BatchNorm на GroupNorm в DeepLabV3 могла незначительно ухудшить результаты, так как GroupNorm менее оптимальна для небольших мини-батчей.

Несмотря на то, что DeepLabV3 является более современной архитектурой, её применение на узких и ограниченных датасетах, таких как CamVid, приводит к ухудшению качества сегментации. FCN демонстрирует более стабильные результаты в подобных условиях и может быть рекомендована для использования в случаях, когда размер данных или вычислительные ресурсы ограничены.

VII. Заключение

Данное исследование было направлено на изучение применимости двух современных нейронных сетей FCN и DeepLabV3 в задаче семантической сегментации объектов на улицах с использованием небольшого датасета CamVid. Основной целью являлось выявление сильных и слабых сторон каждой архитектуры, а также их применимость в условиях ограниченных ресурсов.

CamVid представляет компактный набор данных, содержащий 32 класса объектов, включая мелкие и трудноразличимые, такие как дорожная разметка, светофоры и столбы. Ограниченность объёма данных добавляла сложности задаче, что делало эксперимент ещё более актуальным для практического применения современных моделей.

Для сравнения архитектур был выбран единообразный подход, включающий масштабирование и нормализацию данных, использование комбинированной функции потерь (Jaccard и Dice), а также оптимизатора Adam. Процесс обучения обеих моделей сопровождался техникой early stopping, что обеспечило объективность сравнения.

Результаты эксперимента продемонстрировали, что FCN уверенно справилась с задачей, превзойдя DeepLabV3 по всем основным метрикам. FCN показала IoU 0.324, Dice 19.006 и Pixel Accuracy 0.841, успешно выделяя такие ключевые объекты, как дороги, здания и транспорт. В то же время, DeepLabV3, несмотря на более современную архитектуру, продемонстрировала менее точные предсказания с IoU 0.200 и Pixel Accuracy 0.758. Основной причиной этого стало её высокое требование к объёму данных и сложности параметров, что в условиях ограниченного датасета привело к ухудшению качества сегментации.

Визуальный анализ результатов подтвердил численные метрики: предсказанные маски от FCN более точно отражали основные объекты сцены, тогда как у DeepLabV3 наблюдались значительные артефакты и потеря мелких деталей. Эти различия подчеркивают, насколько важен выбор архитектуры в зависимости от специфики задачи.

Значимость данного исследования заключается в том, что оно демонстрирует: при работе с ограниченными ресурсами, такими как небольшой объём данных или недостаток вычислительных мощностей, FCN оказывается более эффективным выбором, предлагая сбалансированное сочетание точности и оптимальных требований к ресурсам. Однако, если объём данных будет увеличен, а условия позволят задействовать более мощные модели, DeepLabV3 имеет потенциал для значительного улучшения результатов.

Перспективы дальнейших исследований включают изучение оптимальных методов аугментации данных, например, повышение контрастности для более явного отделения одних объектов от других, расширение обучающей выборки.

Таким образом, проведённая работа показала, что даже в условиях ограниченных ресурсов можно достичь лучших результатов сегментации, если выбрать подходящую архитектуру. FCN доказала свою надёжность и

устойчивость, тогда как DeepLabV3 требует более благоприятных условий для раскрытия своих возможностей.

VIII. ЛИТЕРАТУРА

[1] Long J., Shelhamer E., Darrell T. “Fully Convolutional Networks for Semantic Segmentation”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

[2] Chen L.-C., Papandreou G., Schroff F., Adam H. “Rethinking Atrous Convolution for Semantic Image Segmentation”, arXiv preprint arXiv:1706.05587, 2017. Available at: https://arxiv.org/abs/1706.05587 (Accessed: December 25, 2024).

[3] Brostow G. J., Fauqueur J., Cipolla R. “Semantic Object Classes in Video: A High-Definition Ground Truth Database”, Pattern Recognition Letters, Volume 30, Issue 2, 2009, pp. 88–97.

[4] Paszke A., Gross S., Massa F., et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 8024–8037.

[5] Albumentations Library Documentation. Available at: https://albumentations.ai (Accessed: December 25, 2024).

[6] CamVid Dataset. Available at: http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid (Accessed: December 25, 2024).

[7] Garcia-Garcia A., Orts-Escolano S., Oprea S., et al. “A Review on Deep Learning Techniques Applied to Semantic Segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018, pp. 2483–2501.

[8] Zhao H., Shi J., Qi X., et al. “Pyramid Scene Parsing Network”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2881–2890.

[9] Badrinarayanan V., Kendall A., Cipolla R. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017, pp. 2481–2495.

[10] He K., Zhang X., Ren S., Sun J. “Deep Residual Learning for Image Recognition”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[11] Lin T.-Y., Maire M., Belongie S., et al. “Microsoft COCO: Common Objects in Context”, European Conference on Computer Vision (ECCV), 2014, pp. 740–755.

[12] Albumentations Library Documentation. Available at: https://albumentations.ai (Accessed: December 25, 2024).

[13] Paszke A., Gross S., Massa F., et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, Advances in Neural Information Processing Systems (NeurIPS), 2019, pp. 8024–8037.

[14] Brostow G. J., Fauqueur J., Cipolla R. “Semantic Object Classes in Video: A High-Definition Ground Truth Database”, Pattern Recognition Letters, Volume 30, Issue 2, 2009, pp. 88–97.

[15] Chen L.-C., Papandreou G., Schroff F., Adam H. “Rethinking Atrous Convolution for Semantic Image Segmentation”, arXiv preprint arXiv:1706.05587, 2017. Available at: https://arxiv.org/abs/1706.05587 (Accessed: December 25, 2024).

[16] Long J., Shelhamer E., Darrell T. “Fully Convolutional Networks for Semantic Segmentation”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440.

[17] He K., Zhang X., Ren S., Sun J. “Deep Residual Learning for Image Recognition”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[18] Леонов, И. Ю. Классификация транспортных средств компьютерным зрением / И. Ю. Леонов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 46-52. – EDN JSRESQ.

[19] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems", 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-5,. DOI: 10.23919/ICINS51816.2023.10168407

[20] li, Bushra & Sadekov, Rinat. (2023). A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems. Gyroscopy and Navigation. 30. 87-105. DOI: 10.17285/0869-7035.00105

[21] Кирвяков, В. О. Исследование возможности детектирования дорожных знаков / В. О. Кирвяков // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 145-150. – EDN YUWXWU.

ГРНТИ 28.23.37
ASJC 1707
OECD 1.02.EP

Использование подходов семантической сегментации и детектирования в задаче распознавания лавин на изображениях

В. В. Ащепкова
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2411823@edu.misis.ru

Г. С. Листратенков
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2415186@edu.misis.ru

***Аннотация* — в рамках данной работы проводится сравнительный анализ двух подходов к решению задачи распознавания лавин на изображениях: детектирования с использованием модели YOLOv8 и семантической сегментации с применением архитектуры U-Net. Эксперименты выполнялись на специализированных наборах данных с целью оценки производительности методов в условиях различных типов изображений. Показатели точности (mAP, IoU) демонстрируют преимущества и ограничения каждого подхода, что позволяет сделать выводы о целесообразности их использования в зависимости от конкретных условий задачи.**

Ключевые слова — компьютерное зрение, глубокое обучение, распознавание лавин, семантическая сегментация, детектирование, YOLOv8, U-Net, mAP, IoU, безопасность в горных районах.

I. Введение

Изучение и проектирование систем автоматического распознавания природных явлений, таких как лавины, представляет собой важное направление в области обеспечения безопасности и мониторинга окружающей среды [1]. Лавины являются значимой угрозой в горных районах, и своевременное выявление лавинных зон позволяет минимизировать человеческие и экономические потери. С середины 2000-х годов многие университеты, научно-исследовательские центры и компании активно разрабатывают технологии автоматической обработки изображений для идентификации лавин, на международных конференциях, таких как "Гео-Сибирь", обсуждаются методы и алгоритмы обработки спутниковых данных, которые могут быть применены для мониторинга лавин [2-4], в Национальной академии наук Беларуси отмечается активное развитие геосервисов, где данные дистанционного зондирования Земли используются для различных тематических слоёв, включая потенциально и для анализа лавинной активности [5].

Важной задачей при создании подобных систем является точное и надежное распознавание лавинных зон на основе изображений. Для решения этой задачи многими авторами рассматривается вопрос применения компьютерного зрения, в частности методов глубокого обучения. Среди них можно выделить два основных подхода: детектирование объектов [6] и семантическая сегментация [7]. Детектирование объектов позволяет локализовать лавины на изображении в виде ограничивающих рамок, тогда как семантическая сегментация предоставляет детальную информацию о форме и размерах лавин, что особенно важно для анализа и прогноза их последствий.

Современные модели глубокого обучения, такие как YOLOv8 [8] и U-Net [9], зарекомендовали себя как мощные инструменты в задачах классификации, детектирования и сегментации изображений. YOLOv8 (You Only Look Once) представляет собой одну из передовых архитектур для детектирования объектов, обеспечивающую высокую производительность и точность. U-Net, в свою очередь, является классической архитектурой для семантической сегментации, которая демонстрирует превосходные результаты при анализе сложных визуальных данных.

Для обучения моделей глубокого обучения требуются большие объемы данных, что может быть вызовом при работе с природными явлениями, такими как лавины. Однако благодаря доступности специализированных ресурсов по мониторингу лавин, например, таких как: “Gallatin National Forest Avalanche Center” [10], это стало возможным.

В данной работе проводится сравнительный анализ подходов, основанных на детектировании объектов (YOLOv8) и семантической сегментации (U-Net), в задаче распознавания лавин на изображениях. Исследование направлено на оценку применимости методов в различных условиях и анализ их эффективности в реальных сценариях.

II. Наборы данных

Для проведения анализа эффективности двух нейронных сетей: YOLOv8 в задаче детектирования лавины и U-Net в задаче семантической сегментации лавины, были использованы два авторских датасета. Первый - из открытого источника, второй был сформирован авторами для эксперимента.

A. UIBK Avalanche Dataset

Датасет [11] представляет собой обширную коллекцию из 4090 помеченных фотографий, на которых запечатлены различные типы лавин: скользящий, рыхлый, плита. Они соответствуют различным механизмам спуска лавин. Эксперты тщательно аннотировали видимые лавины с помощью полигональных ограничительных коробок и

прямоугольных ограничительных рамок, присваивая каждой из них определенный ярлык.

Состав набора данных:

- изображения:
  - скользящие - 716 фотографий;
  - рыхлые: -416 фотографий;
  - плита - 1 887 фотографий;
  - отсутствуют - 1 071 фотография.
- аннотации:
  - скользящие - 2 489 экземпляров;
  - рыхлые - 1 827 случаев;
  - плита -2 912 случаев.

Фотографии были получены в ходе полевых исследований, проводившихся в течение 21 зимнего сезона, с 2000/2001 по 2021/2022 гг. В названии каждого файла изображения указаны дата и примерное место съемки. Хотя большинство снимков было сделано с земли, некоторые были сняты с низколетящих вертолетов, так как они напоминали ракурсы, достижимые с помощью веб-камер.

Данные содержат изображения в формате “jpg”, отсортированные по общей метке изображения в подгруппы: скользящий, рыхлый, плита, отсутствие лавины , а также аннотации. Для каждой подгруппы они хранятся в формате “json” для каждого изображения. В них записаны полигональные и прямоугольные ограничительные рамки лавины.

Рисунок 1. Изображения датасета UIBK Avalanche Dataset

B. Avalanche

Для создания данного датасета были подобраны изображения сухих и мокрых лавин из открытых источников в сети интернет. Основой для данного датасета послужили реальные фотографии лавин, сделанные людьми в штатах Монтана, Вайоминг и Айдахо США в период с 2021 по 2024 год.

Состав набора данных:

- скользящие -90,
- рыхлые -110,
- плита - 100.

Всего 300 изображений лавин. Ниже приведены примеры изображений:

Рисунок 2. Изображения датасета Avalanche

III. Подготовка данных

Для проведения эксперимента данные из обоих датасетов были приведены в единый формат и размер — 256x256 пикселей, а также созданы аннотации: Bounding Box для задачи детектирования и маски для задачи семантической сегментации. Данные были поделены на тренировочную и валидационную выборки.

Отдельно для расширения тренировочного и валидационного наборов данных было решено применить аугментацию.

Тренировочный:

- горизонтальный поворот, с вероятностью 0,8;
- метод, с вероятностью 0,7:
  - увеличение резкости;
  - размытие;
  - гауссовский шум.
- метод, с вероятностью 0,7:
  - яркость и контрастность;
  - оттенок и насыщенность;
  - цветовой баланс.

Валидационный:

- горизонтальный поворот, с вероятностью 0,8.

Для обучения нейросетей на датасете Avalanche, разметка была проведена посредством электронного сервиса CVAT.ai. Примеры разметки изображений приведены на рисунке 3.

Рисунок 3 Разметка изображения а) исходное б) с маской в) с границами

Для проведения тестирования эффективности нейронных сетей из датасета UIBK Avalanche Dataset была выделена тестовая выборка из случайных изображений.

IV. Нейросетевые архитектуры

A. YOLOv8

YOLO — это алгоритм, который напрямую классифицирует объект за один проход, используя только одну нейронную сеть для предсказания границ и

вероятности класса, используя полное изображение в качестве входных данных.

Архитектура (см. рис. 4) состоит из позвоночника (backbone), шеи (neck) и головы (head). Backbone представляет собой предварительно обученную конволюционную нейронную сеть (CNN) [12], которая извлекает из входного изображения карты признаков низкого, среднего и высокого уровня. Neck объединяет карты признаков с помощью блоков агрегации путей, таких как Feature Pyramid Network (FPN) [13]. Она передает их в head, которая классифицирует объекты и предсказывает ограничительные рамки.

Head может состоять из одноэтапных или плотных моделей предсказания, таких как YOLO или Single-shot Detector (SSD) [14]. В качестве альтернативы в ней могут использоваться двухэтапные или разреженные алгоритмы предсказания, такие как серия R-CNN [15].

YOLOv8 включает в себя увеличение мозаичных данных, обнаружение без якорей, модуль C2f, отсоединенную head и модифицированную функцию потерь.

Мозаичное дополнение данных смешивает четыре изображения, чтобы предоставить модели лучшую контекстную информацию, до достижения последних десяти эпох обучения.

При безъякорном обнаружении модель напрямую предсказывает среднюю точку объекта и сокращает количество предсказаний ограничительных рамок. Это помогает ускорить Non-max Suppression (NMS) — этап предварительной обработки, на котором отбрасываются неверные предсказания.

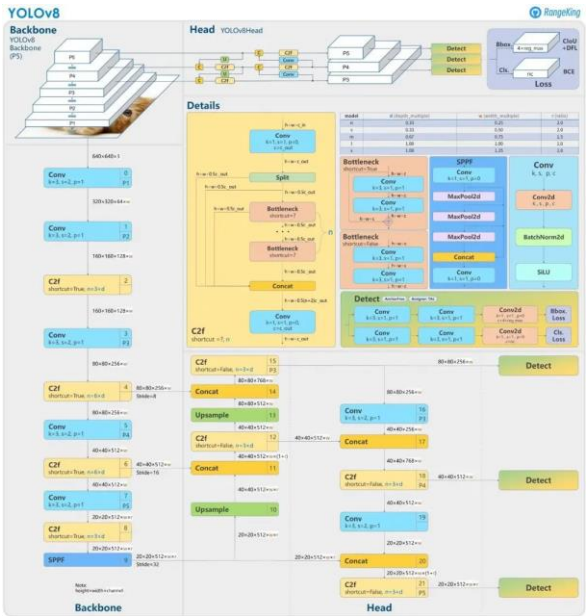


Рисунок 4. Архитектура YOLOv8

Основа модели состоит из модуля C2f вместо модуля C3. Разница между ними заключается в том, что в C2f модель объединяет выходные данные всех узких модулей. В C3 модель использует выход последнего узкого места.

Head выполняет классификацию и регрессию по отдельности.

Основные компоненты функции потерь:

- Оценка согласованности задачи (Task Alignment Score). Этот показатель учитывает как классификацию, так и точность предсказания ограничивающего прямоугольника. Оценка согласованности рассчитывается как произведение классификационного балла и Intersection over Union (IoU). IoU измеряет точность предсказанного прямоугольника (bounding box), показывая, насколько хорошо предсказанный прямоугольник перекрывает истинный (ground truth) прямоугольник.

- Классификационная потеря (Classification Loss). После вычисления оценки согласованности задачи модель выбирает top-k положительных примеров для вычисления классификационной потери. Для классификации используется BCE (binary cross-entropy) loss, которая измеряет разницу между предсказанными и реальными метками классов. Это стандартная метрика для бинарной классификации.

- Потери регрессии (Regression Loss). CIoU (Complete Intersection over Union) Loss. Этот критерий оценивает, как хорошо предсказанный прямоугольник совпадает с истинным, учитывая не только перекрытие (IoU), но и центр предсказанного прямоугольника, соотношение сторон и угол поворота. CIoU даёт более точную метрику для оценки качества локализации. Distributional Focal Loss (DFL). Эта потеря оптимизирует распределение границ предсказанных ограничивающих прямоугольников. Она делает акцент на неправильно классифицированных примерах, особенно на тех, которые модель неверно классифицирует как ложные отрицания (false negatives). Это позволяет модели улучшать точность предсказаний для трудных примеров, на которых она ошибается.

Параметры YOLOv8 включают размеры модели (n, s, m, l, x), подходящие для различных вычислительных ресурсов, и размеры входных данных (стандартный размер — 640x640 пикселей).

Для решения задачи детекции лавин была выбрана модель YOLOv8m, размер картинок стандартный, так как в UIBK Avalanche Dataset хранятся разрозненные данные.

Используемые метрики:

- mAP (mean Average Precision) — средняя точность, которая является основной метрикой для оценки качества модели в задачах детекции объектов;

- FPS (Frames Per Second) — оценка производительности модели, то есть сколько кадров модель способна обработать в секунду;

- IoU (Intersection over Union) — метрическая оценка для оценки того, насколько хорошо предсказанные рамки пересекаются с реальными объектами.

Показатели производительности:

Размер	mAPval	Скорость	Скорость A100	Params	FLOPs
640	50.2	237,4	1,83	25,9	78,9

Таблица 1. Показатели производительности YOLOv8m

B. U-Net

Для задачи семантической сегментации нами была выбрана нейронная сеть U- Net. Это архитектура нейронной сети, специально разработанная для задач сегментации изображений, особенно в медицинской визуализации. Она была представлена в 2015 году исследователями из Университета Фрайбурга и быстро стала одной из самых популярных моделей для сегментации благодаря своей эффективности и точности [16].

Архитектура U-Net имеет форму буквы "U", что отражает ее структуру, состоящую из трех основных частей: энкодера (кодировщика), декодера (декодирующей сети) и пропускных соединений. Энкодер отвечает за извлечение признаков из входного изображения, декодер восстанавливает пространственную информацию для создания сегментированной маски, а пропускные соединения соединяют соответствующие слои энкодера и декодера.

Энкодер состоит из нескольких сверточных слоев и операций подвыборки (пулинга), которые уменьшают пространственные размеры изображения, одновременно увеличивая количество каналов. Это позволяет модели извлекать высокоуровневые признаки.

Декодер состоит из операций увеличения (транспонированная свертка) и сверточных слоев, которые восстанавливают исходное разрешение изображения. Декодер принимает выходы из энкодера и создает финальную маску сегментации.

Одной из ключевых особенностей U-Net являются пропускные соединения, которые соединяют соответствующие слои энкодера и декодера. Это позволяет передавать детальную пространственную информацию, что существенно улучшает качество сегментации, особенно для мелких объектов.

Также для обучения U-Net используется функция потерь, такая как бинарная кросс-энтропия или Dice coefficient, что позволяет эффективно оптимизировать модель для задач сегментации.

Для нашего проекта был выбран энкодер Res-Net-18, это версия энкодера с 18 слоями, которая хорошо подходит для извлечения признаков из изображения. Архитектура сети U-Net с энкодером Res-Net-18 представлена на рисунке:

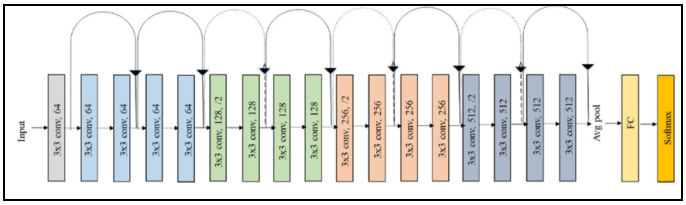


Рисунок 5. Архитектура Res-Net-18

В качестве начальных весов для энкодера будут инициализированы предварительно обученные веса, полученные на наборе данных ImageNet. Это позволит ускорить обучение и повысить производительность модели, т. к. модель уже будет иметь общее представление о признаках изображений, извлеченных из большого и разнообразного набора данных.

В качестве функции активации для выходного слоя будем использовать функцию “softmax2d”, которая применяется к двумерным данным и активно используется в задачах сегментации.

Данная функция преобразует выходные значения нейронной сети в вероятности для каждого класса, что позволяет интерпретировать результаты, как вероятности принадлежности каждого пикселя к определенному классу, В контексте задачи сегментации это позволит модели предсказывать, к какому классу относится каждый пиксель изображения. Для двумерного массива Z размером H×W×C(где H — высота, W — ширина, C — количество классов),функция потерь “softmax2d” определятся как:

$$softmax\left(Z_{i,j,k}\right) = \frac{e^{Z_{i,j,k}}}{\sum_{c=1}^{C} e^{Z_{i,j,c}}},$$

где $e^{Z_{i,j,k}}$ - логит(ненормализованное предсказание) для пикселя (i,j) и класса k. Сумма в знаменателе берется по всем классам c для каждого пикселя (i,j).

Альтернативная функция активации — сигмоида. Данная функция активации широко используется при обучении нейронных сетей, построенных на архитектуре U-Net. Особенно хорошо она проявляет себя в задаче бинарной сегментации. Функция позволяет преобразовать выходные значения в вероятности, что позволяет более эффективно интерпретировать результаты модели. Функция определяется как:

$$\sigma(x) = \frac{1}{1 + e^{-x}},$$

где $\sigma(x)$ — это выход функции активации (в диапазоне от 0 до 1), x — это входное значение (линейная комбинация весов).

Для использования в процессе обратного распространения ошибки (backpropagation) важна также производная функции активации сигмоиды. Она может быть выражена через саму сигмоиду:

$$\sigma'(x) = \sigma(x) * (1 - \sigma(x))$$

Для обучения U-Net будем использовать следующую функцию потерь — “Dice Loss”. Dice Loss основан на коэффициенте Dice, который используется для оценки схожести между двумя наборами данных, в основном, в задачах сегментации изображений. Эта функция потерь особенно полезна, когда классы (например, объекты и фон) имеют значительный дисбаланс, как в нашем случае, больше пространства на изображении занимает фон, при отдаленной съемке, или больше пространства занимает лавина, при близкой съемке. “Dice Loss” определяется как:

$$Dice\ Loss = 1 - Dice\ Coefficient,$$

$$Dice\ Coefficient = \frac{2|X \cap Y|}{|X| + |Y|},$$

где X — предсказанная маска, Y — истинная маска.

Для отслеживания эффективности обучения и работы сети U-Net используются следующие метрики:

1. F1-score.
2. IoU.

Данные метрики активно применяют при задачах сегментации. Рассмотрим каждую метрику подробнее.

F1-score — это гармоническое среднее между точностью (precision) и полнотой (recall). Эта метрика особенно полезна в задачах, где важно учитывать как ложноположительные, так и ложноотрицательные результаты, например, в задачах классификации с дисбалансом классов. Данная метрика определяется как:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall},$$

$$Precision = \frac{TP}{TP + FP} \; Recall = \frac{TP}{TP + FN},$$

где TP — количество истинно положительных предсказаний (True Positives), FP — количество ложноположительных предсказаний (False Positives), FN — количество ложноотрицательных предсказаний (False Negatives).

F1-score принимает значения от 0 до 1, где 1 означает идеальное соответствие (максимальная точность и полнота), а 0 — отсутствие соответствия.

IoU — коэффициент пересечения и объединения, который показывает, как пересечение между предсказанной областью и истинной областью относится к их объединению. Вычисляется по формуле:

$$IoU = \frac{|A \cap B|}{|A \cup B|},$$

где A — область, предсказанная моделью, B — истинная область

Метрика IoU лежит в диапазоне [0, 1] и чем больше ее значение, тем сильнее совпадают предсказанная и истинная маски.

V. Обучение и результаты

A. *YOLOv8*

Для обучения была выбрана модель YOLOv8m. В качестве набора данных был выбран UIBK Avalanche Dataset.

При параметрах:

- количество эпох — 50;
- размер батча — 16;
- оптимизатор — Адам;
- lr — 0,002;
- размер картинок — 640.

Во время первой итерации обучения удалось достичь следующих результатов, которые представлены на рисунке 6 и в таблице 2.

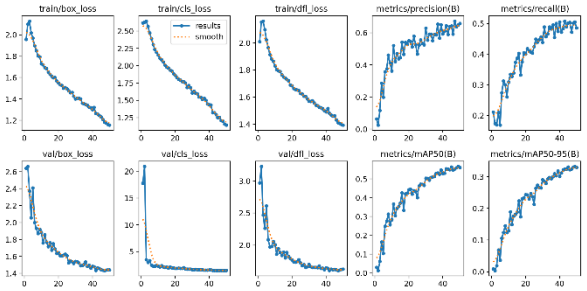


Рисунок 6. Метрики первой итерации обучения

R	mAP50	mAP50-95
0,505	0,567	0,332

Таблица 2. Результаты первой итерации обучения YOLOv8

Функции потерь плавно снижаются по мере увеличения числа эпох. Это демонстрирует стабильность и эффективность обучения, уменьшение ошибок происходит для всех типов потерь (регрессия ограничивающих рамок, классификация и дистрибутивные потери).

Precision (точность) и Recall (полнота) на обучающем наборе данных постепенно растут, то есть модель становится более уверенной и успешной в правильной классификации объектов, а также в обнаружении большего количества объектов.

Метрики mAP50 и mAP50-95 плавно растут. Это ключевые показатели для задач детекции объектов, показывающие качество предсказаний рамок и классов объектов. Рост этих метрик демонстрирует, что модель хорошо обобщает знания на валидационных данных.

Результаты применения к тестовым данным представлены на рисунке 7. На первом изображении модель успешно определила границы лавины. На втором демонстрируется ошибка детекции: модель определила облако в качестве лавины.

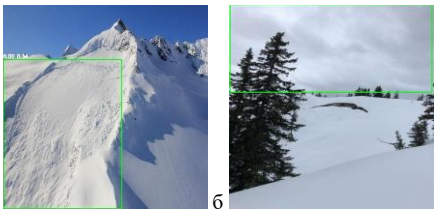


Рисунок 7. Результаты YOLOv8 на тестовых данных

На основе полученных результатов было решено проводить дообучение на датасете Avalanche. Снижена скорость обучения, так как количество данных в датасете кратно снизилось.

Параметры модели:

- количество эпох — 50;
- размер батча — 16;
- оптимизатор — Адам;
- lr — 0,0001;
- размер картинок — 640.

Результаты обучения представлены на рисунке 8 и в таблице 3.

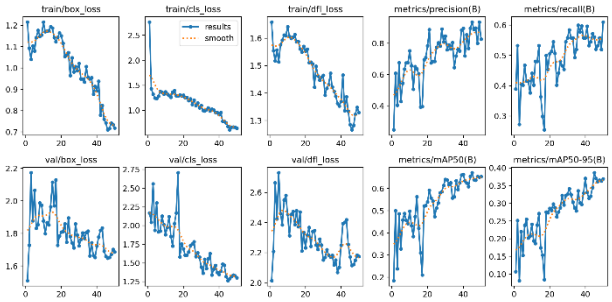


Рисунок 8. Метрики второй итерации обучения

R	mAP50	mAP50-95
0,558	0,671	0,387

Таблица 3. Результаты второй итерации обучения YOLOv8

На рисунке 8 видно, что функции потерь постепенно снижаются. Точность растет по мере обучения, достигая пика и стабилизируясь. Метрики mAP50 и mAP50-95 увеличиваются.

Матрицы показывают, что модель предсказала лавину верно в 60% случаев и ошиблась в 40% случаев. При этом фон определен верно в 100% случаев.

Рассмотрим эффективность модели на тестовых данных (см. рис. 9). На первом изображении определение также успешно, на втором отсутствует ошибка, границы лавины определены точно. Для окончательного подтверждения работоспособности модели нами было взято случайное изображение лавины из сети Интернет, которое не использовалось ни в одном датасете и имеет другой размер. Модель также успешно справилась со своей задачей- границы определены довольно точно.

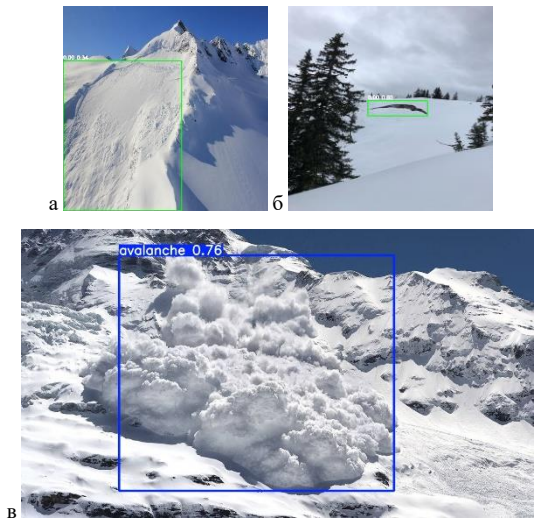


Рисунок 9. Результаты YOLOv8 на тестовых данных а) детекция первого изображения б) улучшенная маска второго изображения в) детекция изображения из Интернета

Исходя из результатов эксперимента, можно сделать вывод, что модель YOLOv8 способна решать задачу детектирования лавин. Для повышения эффективности модели в дальнейшем следует использовать больше изображений с высоким разрешением.

B. U-Net

Первая итерация обучения проходила на данных UIBK Avalanche Dataset.

Параметры обучения:

- количество эпох — 37;
- размер тренировочного батча — 32;
- размер валидационного батча — 16;
- функция активации — softmax;
- функция потерь — DiceLoss.

Метрики обучения представлены на графиках:

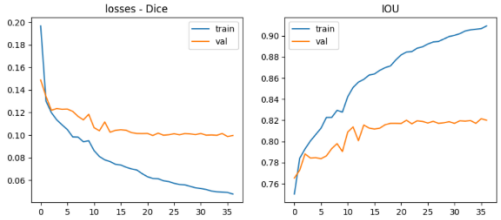


Рисунок 10. Графики метрик обучения U-Net

Потери при обучении уменьшаются с течением эпох. При сравнении с функцией потерь при валидации заметен разрыв между графиками, что может свидетельствовать о чрезмерной подгонке.

На графике IoU видно, что при обучении происходит рост значения метрики с увеличением количества эпох. Что справедливо и для проверки на валидационной выборке. После 25-й эпохи значение вышло на плато, то есть дальнейшее обучение с этими параметрами на этом наборе данных нецелесообразно.

Наилучшие значения для валидационной выборки:

dice_loss	fscore	iou_score
0,1002	0,8998	0,8293

Таблица 4. Результаты первой итерации обучения U-Net

При проверке модели на тестовых данных получены следующие метрики:

dice_loss	fscore	iou_score
0,1015	0,8987	0,8271

Таблица 5. Результаты первой итерации обучения U-Net

При проверке модели на тестовых данных метрики уменьшились незначительно, что говорит о высокой степени определения лавины на новых изображениях. Данные результаты было необходимо подтвердить на практике. Итоговая маска предсказывания представлена на рисунке 10 (б). Из рисунка следует вывод, что маска генерируется неточно, несмотря на высокие показатели метрик.

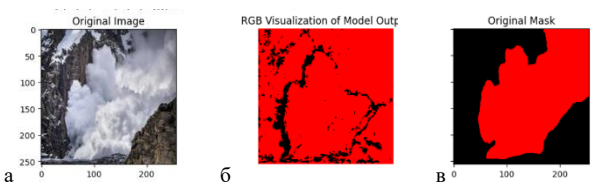


Рисунок 10. Предсказывание лавин после первого обучения U-Net

Учитывая полученные результаты детекции и метрики обучения, проведено дообучение на новых данных с измененными гиперпараметрами. Веса начального состояния модели были взяты из эпохи, при которой было достигнуто максимальное значение IoU.

Далее было проведено несколько итераций обучения на датасете Avalanche для подбора наилучших гиперпараметров модели.

Подобранные параметры обучения:

- количество эпох — 25;
- размер тренировочного батча — 4;
- размер валидационного батча — 1;
- функция активации — сигмоида;
- функция потерь — WeightedDiceLoss, с соотношением значимости классов фона и лавины 1:3.

Метрики обучения представлены на графиках:

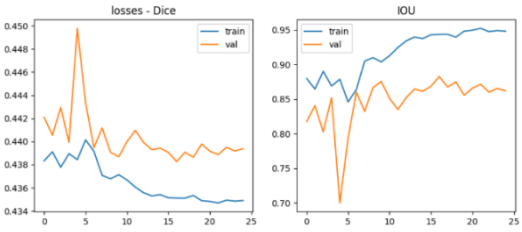


Рисунок 11. Графики метрик обучения U-Net

Проведем анализ графиков. Заметен скачок функции потерь, после которого амплитуда уменьшилась, это говорит о способности модели работать со сложными данными.

Тренировочный IoU демонстрирует устойчивую тенденцию к росту, достигая плато около 18-й эпохи. При этом валидационный показывает некоторые колебания, но остается на более низком уровне

Модель обучается и совершенствуется с течением времени, о чем свидетельствует уменьшение потерь при обучении и увеличение значений IoU.

Существует разрыв между результатами обучения и проверки, что может свидетельствовать о чрезмерной подгонке.

Подобранные нами гиперпараметры обеспечили более высокие показатели на валидационной выборке:

weighted_dice_loss	fscore	iou_score
0,4399	0,9172	0,8519

Таблица 6. Результаты дообучения U-Net

Однако на тестовой выборке модель демонстрирует более низкие показатели точности определения, чем до обучения (см. таб. 7). Это вероятно связано с тем, что для формирования тестовой выборки был использован UIBK Avalanche Dataset, изображения в котором отличаются от данных датасета Avalanche в большей степени, чем друг от друга.

weighted_dice_loss	fscore	iou_score
0,4494	0,8582	0,76

Таблица 7. Результаты дообучения U-Net

Главным критерием эффективности модели является ее работоспособность в реальной ситуации: проведена проверка генерации на изображении для первой итерации (см. рис. 12). Заметно значительное улучшение соответствия предсказанной и истинной масок.

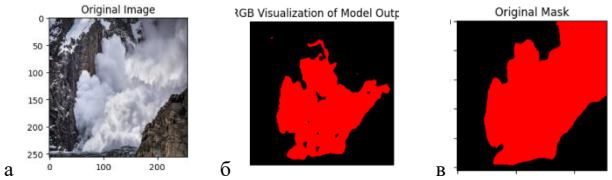


Рисунок 12. Предсказывание лавин после дообучения U-Net

После проведенного эксперимента видно, что нейросеть позволяет распознать лавины и выделить ее область. Изменение гиперпараметров и датасета в процессе обучения позволило повысить эффективность модели.

VI. Заключение

В рамках данной работы был проведен анализ семантической сегментации и детектирования для задачи распознавания лавин на изображении. Нами были подготовлены два датасета для проведения обучения нейронных сетей.

В качестве модели для детекции была выбрана YOLOv8, для семантической сегментации — U-Net. Для каждой из моделей были рассмотрены архитектура, параметры, процесс обучения, тип данных и формат аннотаций. Обучение проводилось в два этапа на разных датасетах, проводился анализ метрик, и проверялась работоспособность модели на тестовых данных.

Модели после дообучения повысили свою точность и стали точнее справляться с задачей распознавания.

Исходя из полученных результатов следует, что YOLOv8 может быть использована в детектировании лавин, а U-Net способен решать задачу их семантической сегментации.

Проведение исследования на двух нейронных сетях, отличных по архитектуре и подходу к обучению, позволило сделать вывод, что YOLOv8 имеет ряд преимуществ: удобство использования, отсутствие необходимости в детальной настройке слоев, низкий порог вхождения. Что касается U-Net, она обладает большим потенциалом: детальная настройка как архитектуры, так и параметров. Нельзя не отметить при этом, что порог вхождения гораздо выше, требуется более детальная работа с данными.

Мы считаем, что для дальнейшего повышения эффективности моделей для задачи распознавания лавин необходимо расширение датасетов, использование изображений высокого разрешения с большим количеством деталей. Для семантической сегментации также следует расширить количество классов (объекты фона: горы, облака, деревья), ограничивающих целевой класс лавин.

VII. Литература

[1] A Survey of Computer Vision Techniques for Forest Characterization and Carbon Monitoring Tasks / S. Illarionova, D. Shadrin, P.

Tregubova [et al.] // Remote Sensing. – 2022. – Vol. 14, No. 22. – P. 5861. – DOI 10.3390/rs14225861. – EDN HXFIQO.

[2] Zhdanov, V.V. (2015). The possibility of using artificial neural networks for avalanche danger forecasting. Geography and Water Resources, 4, 57-62.

[3] UAV Navigation System Autonomous Correction Algorithm Based on Road and River Network Recognition / A. P. Tanchenko, A. M. Fedulin, R. R. Bikmaev, R. N. Sadekov // Gyroscopy and Navigation. – 2020. – Vol. 11, No. 4. – P. 293-299. – DOI 10.1134/S2075108720040100. – EDN QECCWJ.

[4] Geo-Siberia 2007: Geodesy, Geoinformatics, Cartography, Mine Surveying. Volume 1, Part 1. Available at: https://www.geokniga.org/bookfiles/geokniga-geo-sibir2007geodezi-yageoinformatikakartografiyamarksheyderiyatom1chast1.pdf(Accessed: December 25, 2024).

[5] "Report on Research Work (Appendix B)", available at: https://eec.eaeunion.org/upload/iblock/eb5/Otchet_NIR_Pril_B_14_10_2020_Full.pdf (Accessed: December 25, 2024).

[6] Дедов, А. Д. Обнаружение кораблей на спутниковых изображениях с использованием компьютерного зрения / А. Д. Дедов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 36-41. – EDN RVELMU.

[7] Абакумов, А. А. Вопросы сегментации дорожного слоя / А. А. Абакумов, В. О. Хуако // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 40-45. – EDN UWTAKJ.

[8] "YOLOv8 Models Documentation", available at: https://github.com/ultralytics/ultralytics/blob/main/docs/en/models/yolov8.md (Accessed: December 25, 2024).

[9] "U-Net: Convolutional Networks for Biomedical Image Segmentation", available at: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/ (Accessed: December 25, 2024).

[10] “Gallatin National Forest Avalanche Center”, available at: https://www.mtavalanche.com/ (Accessed: December 25, 2024).

[11] “UIBK Avalanche Dataset”, available at: https://www.uibk.ac.at/geographie/lidar/data/ (Accessed: December 25, 2024).

[12] “Convolutional Neural Networks (CNNs)”, available at: https://www.analyticsvidhya.com/blog/2021/08/convolutional-neural-networks-a-beginners-guide/ (Accessed: December 25, 2024).

[13] “Feature Pyramid Network (FPN)”, available at: https://arxiv.org/abs/1612.03144 (Accessed: December 25, 2024).

[14] “Single-Shot MultiBox Detector (SSD)”, available at: https://arxiv.org/abs/1512.02325 (Accessed: December 25, 2024).

[15] “R-CNN Series”, available at: https://arxiv.org/abs/1311.2524 (Accessed: December 25, 2024).

[16] Ronneberger, O. (n.d.). U-Net: Convolutional Networks for Biomedical Image Segmentation. Available at: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/ (Accessed: December 25, 2024).

[17] “Fueling the AI transformation: Four key actions powering widespread value from AI, right now. Deloitte’s State of AI in the Enterprise, 5th Edition report. October 2022”. available at: https://www2.deloitte.com/content/dam/Deloitte/us/Documents/deloitte-analytics/us-ai-institute-state-of-ai-fifth-edition.pdf (Accessed: December 25, 2024).

[18] Lee, T.; Singh, V. P.; Cho, K. H. (2021). Deep Learning for Hydrometeorology and Environmental Science. Book series: Water Science and Technology Library, volume 99, 204 p.

[19] “Artificial Intelligence in Meteorology Industry 2022. Azati Company. 19 December 2022”. available at: https://azati.ai/artificial-intelligence-in-meteorology/?ysclid=m09woxlitg155305830 (Accessed: December 25, 2024).

[20] “Artificial Intelligence applied to weather forecasting”. available at: https://ict.moscow/case/af7945dacf2b637c18d37470/?ysclid=m09v5m8q1c177429719/ (Accessed: December 25, 2024).

[21] Zhang, Y.; Ngo, H.; Zhang, Y.; Yusof, N.; Wang, X. (2024). Imaging Segmentation of Brain Tumors Based on the Modified U-net Method. Information Technology and Control. 53. 1074-1087. 10.5755/j01.itc.53.4.37719.

ГРНТИ 28.23.37, 28.23.15, 68.39.17
ASJC 1702, 1707
OECD 1.02 EP

Нейросетевые методы идентификации конкретного представителя семейства кошачьих

Е. А. Ашманова
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1805775@edu.misis.ru

И. А. Ширеторова
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2008125@edu.misis.ru

Аннотация — В данной работе рассматривается задача идентификации конкретного представителя пароды Мейн-кун в домашней обстановке. Эта задача находит применение в таких областях, как экология, зоология и мониторинг домашних животных. В рамках данного исследования было проведено изучение эффективности моделей, основанных на нейронных сетях – YOLOv8, Faster R-CNN и DETR– в контексте задачи детекции и идентификации животных. Для обучения и тестирования модели применяются изображения, собранные с использованием камеры. Проводится анализ возможностей разработанных нейросетей, а также оценка и сравнение их эффективности в контексте данной задачи.

Ключевые слова — Нейронные сети, Детекция особей, Распознавание животных, Классификация, YOLO, Faster R-CNN, DETR.

I. Введение

В последние годы наука значительно продвинулась в задаче распознавания самых разных объектов в связи с развитием глубокого обучения [1, 2, 3]. В современных исследованиях и практических приложениях все чаще возникает потребность в использовании технологий искусственного интеллекта для мониторинга и анализа поведения животных [4]. Одной из таких задач является идентификация конкретного представителя определённой группы животных, имеющих схожие внешние признаки. В контексте домашних животных, например, кошек, это может быть полезно для изучения их поведенческих особенностей, мониторинга состояния здоровья, а также обеспечения безопасности.

Для решения данной задачи требуются высокоточные алгоритмы, способные идентифицировать конкретного представителя среди других, схожих по внешним признакам. Традиционные методы анализа изображений зачастую оказываются недостаточно эффективными для решения подобных задач, поскольку они не способны в полной мере учесть сложные и многообразные особенности внешнего вида животных, а также вариативность его поз. Современные нейронные сети, основанные на методах глубокого обучения, предлагают эффективные решения за счет своей способности извлекать и обрабатывать сложные визуальные паттерны.

Ранее предложенные подходы, такие как использование сверточных нейронных сетей для классификации объектов [4] и трансформеров для анализа изображений [5], продемонстрировали высокую эффективность в различных задачах компьютерного зрения. Также активно изучаются специализированные методы идентификации животных, включая модели для анализа изображений в зоологии [6] и индивидуального распознавания [7].

Для идентификации домашних животных, таких как кошки и собаки, были разработаны специализированные методы. Например, Kim et al. [8] предложили архитектуру на основе ResNet для распознавания собак, учитывающую индивидуальные черты. Аналогичные подходы могут быть адаптированы для задач мониторинга кошек, с учетом особенностей их внешнего вида и поведения.

II. Наборы данных

Для дообучения и тестирования рассматриваемых в данной работе нейросетей использовался собственный датасет, направленный на то, чтобы модели смогли научить идентифицировать конкретного представителя породы Мейн-кун в домашней обстановке. Датасет состоит из 166 изображений, которые были тщательно отобраны и разделены на две основные категории: изображения конкретного представителя и изображения похожих на него особей.

В качестве форматов разметки были два популярных формата разметки: COCO и YOLO 1.1. Эти форматы обеспечивают эффективную структурированную разметку, необходимую для выполнения задач компьютерного зрения, таких как детекция и идентификация. Каждое изображение было размечено с использованием ограничивающих прямоугольников, которые точно обводят кота на фотографии, а также содержат метку класса. В рамках задачи рассматриваются два класса: “Mars” (целевой объект для идентификации) и “Other cats”.

Все изображения были предварительно обработаны и приведены к единому размеру 640 на 640 пикселей с помощью добавления черных отступов по краям изображений. Этот метод позволил сохранить исходные пропорции изображений, избегая искажений, которые могли бы возникнуть при обычном масштабировании. Черные рамки добавлялись по краям изображений, чтобы заполнить недостающее пространство и достичь заданного размера. Это позволило унифицировать данные и обеспечить корректную работу моделей, минимизировав искажения и потерю информации.

A. Изображения конкретного представителя

В эту категорию вошло 84 фотографии, сделанные автором, которые представляют собой изображения

одного конкретного представителя кошек пароды Мейн-кун, запечатленный в разных позах. Эти снимки обладают различными ракурсами, что позволяет создать точную модель для задач идентификации.

На рисунках 1–3 изображен с разных ракурсов объект, который является целевым для распознавания.

Рис. 1. Фотография целевого объекта в профиль

Рис. 2. Фотография целевого объекта в фас

Рис. 3. Фотография целевого объекта в нестандартном для идентификации полоэении

B. Изображения похожих особей

В эту категорию вошло 82 фотографии представителей кошачьих, найденные в интернете, которые визуально схожи с исследуемым объектом. Большая часть изображений содержит представителей породы Мейн-кун, в то время как на остальных представлены особи других пород, обладающие общими чертами с целевым объектом. На рисунках 4–6 показаны примеры фотографий кошек, схожих по основным признакам с целевым объектом, запечатленных в различных позах и условиях.

Рис. 4. Пример кадра с кошкой в нестандартном для идентификации положении

Рис. 5. Пример кадра с представителем пароды Мейн-кун похожим на целевой объект основными чертами, который был приведен к размеру 640×640 пикселей

Рис. 6. Пример кадра с котом схожим по основным признакам с целевым объектом

III. Нейросетевые архитектуры

A. Faster R-CNN (Region-based Convolutional Neural Network)

Архитектура Faster R-CNN (Region-based Convolutional Neural Network) представляет собой метод для обнаружения объектов в изображениях [9]. Он был представлен Россом Гиршиком на конференции

Computer Vision and Pattern Recognition (CVPR) в 2015 году. Faster RCNN внесла существенный вклад в область обнаружения объектов, показывая высокую точность и эффективность работы.

На рисунке 7 изображена архитектура Faster R-CNN. Она включает в себя несколько ключевых модулей, каждый из которых выполняет специфическую задачу:

- Сверточная нейронная сеть (Backbone). Эта часть используется для извлечения признаков из изображения. Архитектура принимает входное изображение и пропускает его через предварительно обученную сверточную сеть, такую как ResNet, VGG или MobileNet, что позволяет получить карту признаков (feature map). На этой карте содержится информация о высокоуровневых характеристиках изображения, таких как текстуры, формы и контуры объектов [10].
- Сеть генерации областей (Region Proposal Network, RPN). Ключевой инновацией Faster R-CNN является использование RPN, которая заменяет внешние алгоритмы, такие как Selective Search, применявшиеся в более ранних подходах. RPN принимает карту признаков и генерирует регионы интереса (Region Proposals), где с наибольшей вероятностью находятся объекты. Каждый регион определяется ограничивающей рамкой (bounding box), и сеть прогнозирует вероятность наличия объекта в данном регионе. Для реализации RPN используются Anchors — заранее определённые рамки различных размеров и соотношений сторон. Это позволяет покрывать широкий диапазон возможных форм и размеров объектов [9].
- Pooling полей интереса (RoI Pooling/Align) Предложения, сгенерированные RPN, преобразуются в фиксированный размер через процесс RoI Pooling. Это упрощает обработку на последующих этапах. RoI Pooling выделяет регионы с карты признаков, соответствующие предложенным областям, и масштабирует их до фиксированного размера [11].
- Головная сеть (Detection Head). Головная сеть отвечает за окончательную классификацию объектов и уточнение координат рамок: Классификация предсказывает, к какому классу принадлежит объект. Регрессия рамок уточняет местоположение объектов для повышения точности детекции.

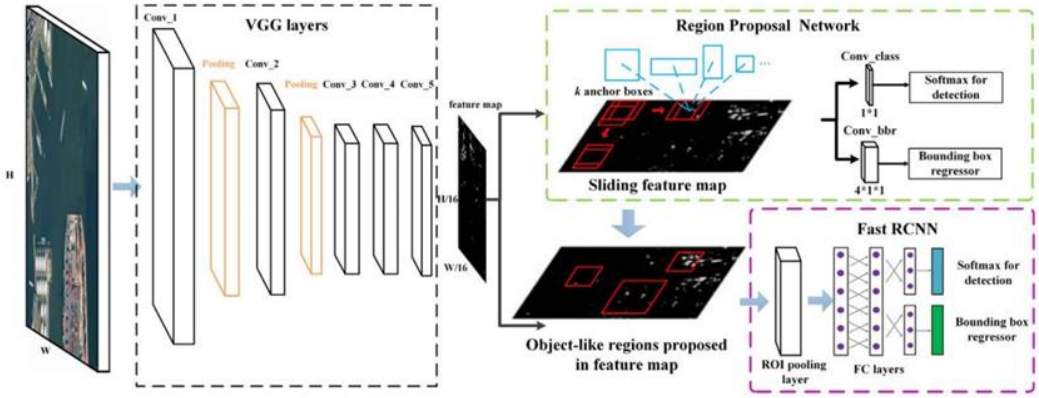


Рис. 7. Архитектура Faster R-CNN на основе VGG

B. *YOLOv8 (You Only Look Once).*

YOLOv8 — это последняя версия популярной нейросетевой архитектуры для детекции объектов в реальном времени, которая продолжает эволюцию оригинальной модели YOLO. YOLOv8 улучшает точность и скорость обнаружения объектов, сохраняя свои ключевые преимущества — скорость работы и универсальность.

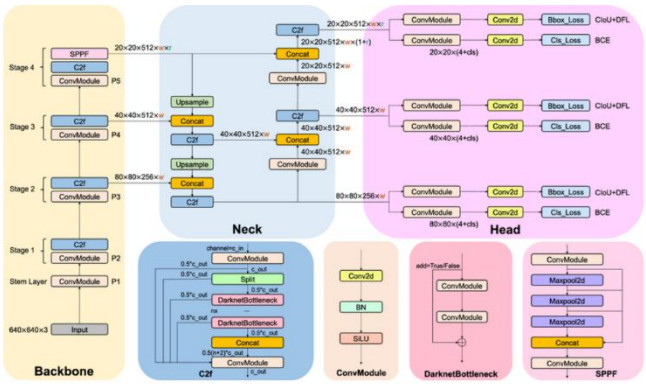


Рис. 8. Схематичное изображение архитектуры YOLOv8

Как и предыдущие версии YOLO, архитектура YOLOv8 использует принцип одноступенчатой детекции, что означает, что модель выполняет как локализацию объектов (определение их положения на изображении), так и классификацию объектов за один проход через нейронную сеть. YOLOv8 строится на

основе более глубокой и сложной сети, что позволяет достичь улучшенной точности при сохранении высокой скорости работы.

Основные компоненты YOLOv8:

- Backbone (основной блок):

  CSPDarknet: для извлечения признаков используется модификация сети Darknet, которая была оптимизирована для более быстрой и эффективной обработки данных. Эта версия использует CSPNet (Cross-Stage Partial Network) для повышения производительности и уменьшения вычислительных затрат [12].

- Neck (средний слой):

  PANet (Path Aggregation Network): Сеть использует PANet для улучшенной агрегации информации с разных уровней признаков. Это позволяет YOLOv8 работать лучше с объектами разного масштаба.

  FPN (Feature Pyramid Network): Эта структура улучшает точность распознавания мелких объектов за счет комбинирования признаков с разных уровней сети [13].

- Head (выходной слой):

  На выходе YOLOv8 предсказывает координаты ограничивающих рамок объектов, вероятность принадлежности объектов к определённым классам, уверенность в каждом предсказании. Также используются методы, такие как IoU (Intersection over Union) для оценки перекрытия рамок и Non-Maximum Suppression (NMS) для удаления лишних предсказаний [14].

YOLOv8 включает в себя несколько усовершенствований, которые помогают улучшить точность и производительность модели. Одним из них является оптимизация вычислений. YOLOv8 использует более компактные и быстрые архитектуры для обработки изображений, что позволяет ускорить обучение и детекцию. Также модель использует многомасштабное обучение, т.е. она обучается с учётом объектов различных размеров, что улучшает её способность работать с малыми и крупными объектами на изображении.

C. DETR (DEtection TRansformers)

DETR — это новаторская архитектура для задачи детекции объектов, основанная на принципах трансформеров. В отличие от традиционных методов, таких как YOLO и Faster R-CNN, которые используют анкоры или схемы регионов для детекции объектов, DETR применяет трансформеры, что позволяет напрямую моделировать взаимодействия между объектами на изображении. Эта архитектура продемонстрировала высокую точность при значительно меньшем количестве гиперпараметров и улучшенной гибкости [15].

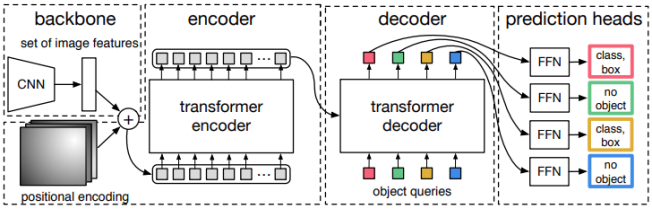


Рис. 9. Схематичное изображение архитектуры DETR

Основные компоненты архитектуры DETR:

- Backbone (Основной блок):

  Для извлечения признаков используется стандартная сверточная нейронная сеть, такая как ResNet, которая преобразует изображение в набор признаков (feature map). Этот этап аналогичен тому, как используется backbone в других архитектурах, таких как Faster R-CNN или YOLO.

- Encoder-Decoder (Трансформер):

  Суть инновации в DETR заключается в использовании трансформеров для обработки признаков, извлечённых из изображения. Архитектура включает два основных компонента: encoder и decoder. Признаки, полученные от backbone, проходят через encoder трансформера. Этот этап обрабатывает все признаки на изображении, запоминая их контекст и взаимосвязи. На выходе энкодера находится набор «сегментированных» признаков, которые затем передаются декодеру трансформера. Декодер генерирует предсказания для каждого объекта, включая координаты ограничивающих рамок и вероятность принадлежности объекта к классу. В отличие от классических подходов, где региональные предложения генерируются заранее, DETR напрямую выдает результаты из трансформера, используя лишь набор фиксированного количества «слотов» (queries).

- Object Queries (Запросы объектов):

  Одной из ключевых особенностей DETR является использование object queries, которые являются фиксированными векторными представлениями объектов на изображении. Каждый запрос представляет собой потенциальный объект, и трансформер обучается связывать эти запросы с реальными объектами, найденными на изображении.

- Heads (Выходные слои):

  После декодирования выходной слой генерирует координаты ограничивающих рамок для каждого объекта и предсказания классов объектов на основе выходных данных. Для улучшения производительности используется Bipartite Matching Loss, который оптимизирует соответствие между реальными объектами и предсказаниями, что минимизирует ошибки.

IV. Сравнение

Сравним три описанных подхода. Было проведено обучение моделей, данные разделены в пропорции

70:15:15 на тренировочную, тестовую и валидационную части соответсвено. Качество работы двух подходов складывается из качества работы, локализующей и классифицирующей частей.

Качество работы модели оценивалось как для локализации объекта, так и для его классификации. Использовались следующие меры:

- TP – модель верно распознала нужного кота
- FP – модель ошибочно определила другого кота за целевой объект.
- FN – Модель не смогла определить целевого кота на изображении, где он есть [16,17].

Стоит отметить, что TN в данном случае не определена, так как это величина означает то, что детектор не определил кота, где его действительно нет. По введённым величинам строятся такие функции оценок, как:

- $Precision = \frac{TP}{TP+FP}$ – сколько раз модель определила кота там, где оно действительно есть к общему числу детектированных котов;
- $Recall = \frac{TP}{TP+FN}$ – сколько котов обнаружила модель от общего числа котов;
- $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$ – оценка баланса между точностью (precision) и полнотой (recall) [18,19].

В таблице 1 показаны количественные характеристики двух используемых подходов [20, 21]. Самые высокие показатели имеет модель YOLO, остальные модели показывают результат значительно хуже. Из-за особенность архитектуры и качества изначальных моделей YOLO показывает наилучшую работоспособность на данной задаче с достаточно маленьким объемом данных для обучения.

ТАБЛИЦА I. Оценка детектирующей части

	YOLOv8	**Faster R-CNN**	**DETR**
Precision	0.85	0.15	0.017
Recall	0.80	0.24	0.053
F1	0.82	0.19	0.026
mAP@50	0.75	0.16	0.005
mAP@50-95	0.68	0.23	0.001

В оценку детектирующей части включены три модели: YOLOv8, Faster R-CNN и DETR. Каждая из них была протестирована на одном и том же наборе данных. Результаты данной работы представлены в Таблице I. Основные метрики, такие как Precision, Recall, F1-мера, mAP@50 и mAP@50-95, позволяют сравнить эффективность моделей в задачах идентификации конкретного представителя кошачьих пароды Мейн-кун.

Рис. 10. Пример корректной идентификации целевого объекта с помощью модели YOLOv8

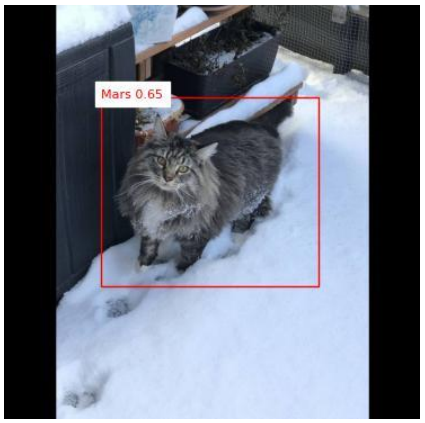


Рис. 12. Пример неверной идентификации другого представителя семейства кошачьих, принятого за целевой объекта моделью YOLO

Рис. 12. Пример неверной идентификации целевого объекта с помощью модели Faster R-CNN

Как видно из таблицы детектор YOLOv8 имеет значительно более высокие показатели, что означает, что он намного лучше справляется с задачей по идентификации конкретного представителя. Модель YOLOv8 демонстрирует высокую точность и полноту, хорошо балансируя между ними.

Сравнивая модели, можно заметить, что Faster R-CNN и DETR показывают значительно более низкие результаты. При тестировании данных моделей было выявлено значительное количество ошибок при детекции животных, на многих картинках они не смогли

определить кота. Если же моделям удавалось определить кота на изображении, то они чаще всего они ошибались с идентификацией целевого объекта, относя его к классу “Other cats”. Это может быть связано с особенностями архитектуры этих моделей. Например, DETR, основанная на трансформерах, может быть менее эффективной на небольших наборах данных.

Во время обучения для моделей Faster R-CNN и DETR был использован ряд стратегий: изменения количества эпох обучения от 10 до 50, а также пробование различных оптимизаторов с настройкой гиперпараметров (таких как скорость обучения, моментам и другие). Однако, максимальный результат производительности, который был получен, оставался относительно скромным, как показано в таблице. Это указывает на возможные ограничения модели при работе с ограниченными данными, где даже с различными настройками гиперпараметров и долгим обучением не удавалось существенно улучшить точность.

Несмотря на теоретические преимущества трансформеров в обработке последовательностей и их успешное применение в различных задачах компьютерного зрения, DETR показал низкую производительность на нашем наборе данных. Это, скорее всего, связано с несколькими факторами, такими как высокая вычислительная нагрузка, медленный процесс обучения и необходимость в большом объеме данных для полноценного функционирования. В результате модель DETR продемонстрировала значительно большую ошибку по сравнению с более легкими и быстрыми подходами. Это указывает на то, что для задач с ограниченными данными и ресурсами трансформеры пока что не являются оптимальным выбором.

Модель YOLO продемонстрировала значительно лучшие результаты на небольшом объеме данных. Несмотря на свою простоту и менее сложную архитектуру по сравнению с Faster R-CNN, YOLO справляется с задачей идентификации гораздо более эффективно. Данный подход оказался особенно хорош в условиях ограниченных данных, что свидетельствует о его способности быстро адаптироваться и показывать высокое качество работы на малых выборках, при этом поддерживая высокую скорость обработки.

V. Заключение

В данной работе были рассмотрены и проанализированы три современные архитектуры нейронных сетей — Faster R-CNN, YOLOv8 и DETR, применяемые для задачи идентификации конкретного представителя породы Мейн-кун в домашних условиях. Было проведено тестирование моделей на заранее размеченном датасете с использованием форматов COCO и YOLO1.1, содержащем фотографии как целевого кота, так и других представителей семейства кошачьих со схожими внешними признаками. Для анализа производительности использовались такие метрики, как Precision, Recall, F1-score, mAP@50 и mAP@50-95 а также значения TP, FP, FN и TN.

В ходе данной работы были получены результаты, которые показывают, что YOLOv8 является наиболее эффективной моделью для задачи идентификации конкретного представителя определённой группы животных, имеющих схожие внешние признаки. Несмотря на небольшой объем датасета данная модель продемонстрировала высокую точность, что связано с её оптимизированной архитектурой, способностью эффективно использовать предобученные веса и адаптироваться к ограниченным данным.

Литература

[1] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," 2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[2] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 10.23919/ICINS51816.2023.10168407. 2023, pp. 1-5.

[3] D. S. Matyash, "Detection of unmanned aerial vehicles in photographs using computer vision techniques", Artificial intelligence in industrial, commercial, medical and financial applications collection of articles of the scientific and technical seminar of students of the department of engineering cybernetics. — 2024: National Research Technological University "MISIS", pp. 92-99.

[4] A.A. Stupina, "Investigation of the possibility of animal recognition in an artificial environment", Artificial intelligence in industrial, commercial, medical and financial applications collection of articles of the scientific and technical seminar of students of the department of engineering cybernetics. — 2023: National Research Technological University "MISIS", pp. 62-67.

[5] A. Krizhevsk, I. Sutskever and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[6] A. Dosovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *International Conference on Learning Representations (ICLR)*, 2021, pp. 1–17.

[7] S. Schneider, G.W Taylor and S.C. Kremer, "Deep Learning for Animal Re-Identification in Wildlife Research," *Nature Communications*, vol. 11, no. 1, 2020, doi: 10.1038/s41467-020-17316-7.

[8] D Deb, "Face Recognition: Primates in the Wild," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4854–4862.

[9] J. Kim, "Deep Learning-Based Pet Identification System Using Convolutional Neural Networks," *Applied Sciences*, vol. 10, no. 4, 2020, pp. 1–14, doi: 10.3390/app10041234..

[10] S. Ren, K. He, R. Girshick and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15, pages 91–99, Cambridge, MA, USA, 2015. MIT

[11] R. Girshick, **Fast R-CNN**. *Proceedings of the IEEE International Conference on Computer Vision*. Available at: https://arxiv.org/abs/1504.08083 (Accessed: December 13, 2024).

[12] J Huang, K. Chen and Z. Liu, (2017). **Speed/accuracy trade-offs for modern convolutional object detectors**. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Available at: https://arxiv.org/abs/1611.10012 (Accessed: December 13, 2024).

[13] J. Redmon, S. Divvala, R. Girshick, R and A. Farhadi, (2016). **You Only Look Once: Unified, Real-Time Object Detection**. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Available at: https://arxiv.org/abs/1506.02640 (Accessed: December 13, 2024).

[14] J. Redmon and A. Farhadi, **YOLO9000: Better, Faster, Stronger**. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Available at: https://arxiv.org/abs/1612.08242 (Accessed: December 13, 2024).

[15] Redmon, J., & Farhadi, A. **YOLOv3: An Incremental Improvement**. *arXiv preprint*. Available at: https://arxiv.org/abs/1804.02767 (Accessed: December 13, 2024).

[16] Carion, N., et al. . **End-to-End Object Detection with Transformers**. *arXiv preprint*. Available at: https://arxiv.org/abs/2005.12872 (Accessed: December 13, 2024).

[17] Keylabs, "Confusion Matrix: TP, FP, FN, TN explained," *Keylabs Blog*, https://keylabs.ai/confusion-matrix-tp-fp-fn-tn-explained (Accessed: December 13, 2024).

[18] Evidently AI, "How to interpret a confusion matrix for a machine learning model," *Evidently AI Blog*, https://www.evidentlyai.com/blog/confusion-matrix (Accessed: December 13, 2024).

[19] Analytics Vidhya, "Precision, Recall, and F1 Score Explained," *Analytics Vidhya Blog*, https://www.analyticsvidhya.com/blog/2020/04/precision-recall-and-f1-score/ (Accessed: December 13, 2024).

[20] Towards Data Science, "Confusion Matrix and Performance Metrics," *Towards Data Science Blog*, https://towardsdatascience.com/an-introduction-to-performance-metrics-in-machine-learning-543bfa9256b1 (Accessed: December 13, 2024).

[21] Z. C. Lipton, C. P.Elkan, B. Narayanaswamy. “Thresholding Classifiers to Maximize F1 Score”, 2014 arXiv: Machine Learning, pp. 1-16.

[22] M. Sokolova, N. Japkowicz, S. Szpakowicz. “Beyond accuracy, Fscore and ROC: a family of discriminant measures for performance evaluation”, Proceedings of Australasian joint conferenceon artificial intelligence, 2006, vol. 4304, pp. 1015-1021.

ГРНТИ 28.23.15
ASJC 1707
OECD 01.02.EP

Исследование возможности детектирования дорожных знаков на основе нейросетевой модели YOLO

Ф. Е. Базалеев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2314593@edu.misis.ru

Е. И. Пиховская
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2409987@edu.misis.ru

***Аннотация* — в данной статье проводится сравнительный анализ двух версий нейронных сетей YOLO — YOLOv8 и YOLOv5 — для задачи детектирования дорожных знаков, связанных с железнодорожными переездами. Обучение моделей проводилось на уникальном наборе данных, включающем три типа автодорожных знаков, размеченных вручную в системе CVAT. Для оценки моделей использовались стандартные метрики. Проведённый анализ показал, что YOLOv8 превосходит YOLOv5 по большинству ключевых показателей точности, однако YOLOv5 демонстрирует более стабильное время обработки, что делает её эффективной для систем реального времени. Полученные результаты могут быть полезны для разработки решений в области автономного управления транспортными средствами и систем помощи водителям.**

Ключевые слова — детектирование, распознавание дорожных знаков, нейронные сети, железнодорожный переезд, YOLO.

I. Введение

Дорожные знаки играют ключевую роль в обеспечении безопасности дорожного движения, предоставляя водителям и пешеходам необходимую информацию для принятия решений. Развитие технологий компьютерного зрения и искусственного интеллекта открывает новые возможности в области обеспечения безопасности дорожного движения. Технологии автоматического детектирования и распознавания дорожных знаков становятся всё более актуальными с развитием автономных транспортных средств и систем помощи водителям (ADAS) [1][2]. Эти системы требуют точной и надежной работы алгоритмов компьютерного зрения, чтобы эффективно идентифицировать знаки в реальном времени, независимо от условий освещения, погоды или иных факторов, влияющих на видимость.

Как отмечается в исследованиях [3][4], алгоритмы компьютерного зрения становятся ключевым инструментом для автономных систем, обеспечивая их способность адаптироваться к сложным условиям окружающей среды. Современные нейросетевые архитектуры, такие как YOLO (You Only Look Once), зарекомендовали себя как эффективные инструменты для решения задач детектирования объектов на изображениях.

В данной статье рассматривается исследование возможностей детектирования автомобильных знаков, связанных с железнодорожными переездами. Для обучения и сравнения производительности использовались две версии модели YOLO — YOLOv8 и YOLOv5. В работе описан набор данных, который включает изображения с тремя типами автодорожных знаков, обозначающих железнодорожные переезды. Также подробно рассмотрены архитектурные особенности используемых моделей и их влияние на качество обучения.

Основной целью исследования является сравнение результатов работы двух моделей по метрикам точности, полноты, времени инференса и другим показателям.

II. Наборы данных

Для создания и обучения моделей был использован уникальный датасет, составленный из изображений двух различных источников, что обеспечило разнообразие данных и их соответствие поставленной задаче. Датасет составлялся из знаков, связанных с жд переездами:

- 1.1 - железнодорожный переезд со шлагбаумом;
- 1.2 - железнодорожный переезд без шлагбаума;
- 1.3.1 - однопутная железная дорога.

Источники датасета:

1. Russian Traffic Sign Dataset (RTSD)

Russian Traffic Sign Dataset (RTSD) — это открытый набор данных, предназначенный для задач компьютерного зрения, связанных с распознаванием и детектированием дорожных знаков. Датасет ориентирован на дорожные условия России и включает изображения с дорожными знаками, собранными в различных климатических и погодных условиях, что делает его особенно ценным для задач, связанных с автоматизацией управления транспортными средствами в российских условиях.

Из этого набора данных было выбрано 400 изображений. Однако оригинальная разметка знаков, представленная в RTSD, не подошла для ~~нашей~~ задачи, так как она либо не содержала нужных классов, либо имела недостаточную точность. Поэтому все изображения из данного набора были размечены вручную с использованием платформы CVAT.

Рис. 1. Примеры размеченных кадров из датасета RTSD

Датасет RTSD состоит из большого количества изображений различного качества и погодных условий, что позволит тренировать модель в реальных условиях. Это особенно важно в контексте задач, связанных с автономным управлением транспортных средств, где модели должны учитывать сложные условия, такие как узкие дороги, ограниченная видимость и плотная городская застройка, что отмечается в исследовании [5].

Рис. 2. Примеры кадров из датасета RTSD, снятых в сложных условиях обстановки

2. Скриншоты карт

Вторым источником данных стали изображения, собранные самостоятельно с использованием скриншотов карт. Для этого было получено 288 изображений различных дорожных участков, на которых встречаются железнодорожные переезды. Эти изображения были обработаны аналогичным образом: вручную размечены в CVAT. Данный подход позволил включить в датасет редкие и специфические случаи, которые не всегда представлены в открытых наборах данных, а также адаптировать выборку к реальным условиям, характерным для использования моделей в российских регионах.

Собственная разметка и выбор датасета позволили найти особенные случай (перекрытие одним знаком другого, нахождение большого количества знаков в одно месте и др), которые позволят модели лучше адаптироваться к реальным условиям.

Рис. 3. Примеры снимков с частично перекрытыми знаками

Общий объем датасета составил 688 изображений, что обеспечило достаточный объем данных для обучения современных моделей глубокого обучения Форматом датасета YOLOv8 Detection. Для достижения сбалансированного распределения данных и предотвращения смещения результатов все изображения были разделены на обучающую, тестовую и валидационную выборки в соотношении 70:15:15. Это позволило использовать основную часть данных для оптимизации параметров моделей, а оставшуюся – для независимой оценки их производительности.

Важно отметить, что при составлении датасета был учтен баланс классов. Все изображения содержат хотя бы один из трех целевых дорожных знаков, что гарантировало равномерное представление каждого класса в обучающей выборке. Дополнительно, в рамках подготовки данных, был проведен анализ качества аннотаций, чтобы минимизировать ошибки разметки, которые могли бы негативно повлиять на обучение моделей.

III. Нейросетевые архитектуры

Для решения задачи детекции дорожных знаков были выбраны модели YOLOv5 и YOLOv8. Эти архитектуры представляют собой современные решения для задач компьютерного зрения, обеспечивающие высокую точность и производительность. Их выбор был обусловлен следующими причинами:

- YOLO (You Only Look Once) является одной из самых популярных архитектур для детекции объектов в реальном времени. Начиная с версии YOLOv3 [6].
- YOLOv5 и YOLOv8 включают поддержку пользовательских наборов данных и позволяют гибко адаптировать модель к специфическим классам объектов. Это особенно важно для задачи, где требуется точная детекция трех классов дорожных знаков
- В отличие от других моделей, таких как Faster R-CNN [7], YOLO может демонстрировать хорошую производительность даже на относительно небольших наборах данных за счет эффективного использования предобученных весов и методов аугментации.

Архитектуры YOLOv5 и YOLOv8 используют сверточные нейронные сети для обнаружения объектов и имеют схожие принципы работы, однако различаются по функциональным возможностям и архитектурным решениям.

Основные компоненты архитектуры YOLOv5:

Backbone (извлечение признаков CSPDarknet) YOLOv5 использует модификацию базовой архитектуры Darknet под названием CSPDarknet (Cross-Stage Partial Darknet). Этот компонент отвечает за извлечение признаков из входных изображений. Основная цель CSPDarknet — сохранить высокий уровень производительности при уменьшении вычислительных затрат. Это достигается путем разделения и частичной агрегации слоев для уменьшения избыточности в процессе передачи данных.

Neck (связующий компонент PANet): Для улучшения передачи признаков между слоями используется PANet. Этот блок помогает усиливать особенности объекта на различных уровнях масштабирования, что улучшает точность на изображениях с объектами разных размеров. PANet помогает в выделении более точных признаков, объединяя информацию из разных уровней сети.

Head (выходная часть Predictions): В выходной части YOLOv5 используются якоря (anchor boxes) для предсказания координат bounding box'ов и классификации объектов. Модель предсказывает координаты (x, y), ширину и высоту для каждого обнаруженного объекта, а также его класс. В отличие от предыдущих версий YOLO, YOLOv5 улучшил механизм якорей для более точной локализации объектов.

Метод обучения:

- Auto-Learning Rate: YOLOv5 использует механизм автоматической настройки скорости обучения, что позволяет модели адаптироваться к изменениям данных во время тренировки. Это улучшает качество обучения и сокращает время, необходимое для достижения хороших результатов.
- Data Augmentation: YOLOv5 включает улучшенные методы аугментации данных, такие как случайные обрезки, изменения яркости и контраста, что помогает улучшить обобщающую способность модели.

Оптимизация: YOLOv5 оптимизирован для работы в PyTorch, но также поддерживает экспорт в ONNX, TensorRT и CoreML, что делает его удобным для использования на различных платформах и устройствах.

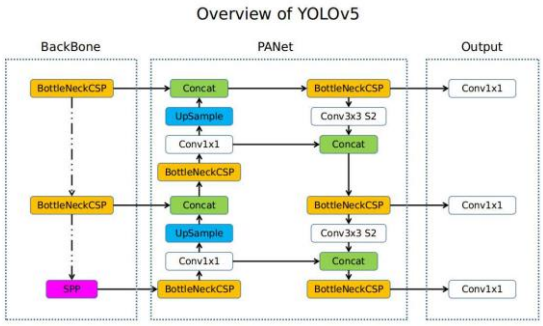


Рис. 4. Архитектура YOLOv5

Основные компоненты архитектуры YOLOv8:

Backbone (извлечение признаков EfficientNet): YOLOv8 использует более современную архитектуру, включающую EfficientNet или другие компоненты, оптимизированные для высокоэффективного извлечения признаков. Эти архитектуры позволяют модели работать быстрее и точнее, особенно на изображениях с большим количеством объектов. EfficientNet использует баланс между глубиной сети, шириной и резолюцией, что значительно улучшает производительность по сравнению с более простыми архитектурами.

Neck (связующий компонент FPN): в YOLOv8 дополнительно используются более сложные методы для усиления передачи признаков между слоями, такие как Feature Pyramid Networks и Self-Attention. С помощью Attention механизмов модель может "фокусироваться" на более важных областях изображения, улучшая точность в обнаружении объектов.

Head (выходная часть Improved Prediction Head): в YOLOv8 выходная часть модели включает улучшенные предсказания для локализации объектов, использующие более сложные методы обработки координат и классов. Также в выходе используется динамическая настройка якорей для повышения точности на разных масштабах объектов.

Мультизадачное обучение (Multitask Learning): поддержка мультизадачности: YOLOv8 имеет улучшенную способность к решению нескольких задач одновременно, таких как обнаружение объектов, сегментация, классификация. Это позволяет использовать одну модель для множества типов задач, делая её более универсальной.

Метод обучения:

- Dynamic Learning Rate Scheduling: в YOLOv8 используется более сложная настройка скорости обучения, которая адаптируется в процессе тренировки, что повышает общую эффективность обучения и улучшает финальный результат.
- Hard Negative Mining: для улучшения работы модели на сложных примерах, YOLOv8 использует метод hard negative mining, который помогает модели учиться на ошибках, фокусируясь на самых трудных примерах.

Оптимизация: YOLOv8 оптимизирован для использования на широком спектре устройств, включая мобильные и встроенные системы. Модель имеет улучшенную поддержку для TensorRT, ONNX и других форматов, что делает её удобной для интеграции в различные платформы и обеспечивает хорошую производительность на ограниченных по мощности устройствах.

IV. Сравнение

Оценка эффективности работы обученных моделей рассчитывается из качества работы, классифицирующей части и оценки локализации, которая производится при помощи расчета индекса Жаккара (Intersection over

Union, IoU) для каждой детекции [8]. В задачах детектирования объектов использовались следующие величины:

- TP (True Positive) — это количество объектов, которые модель правильно обнаружила и классифицировала как целевые.
- FP (False Positive) — это количество объектов, которые модель ошибочно обнаружила как целевые, хотя в действительности они таковыми не являются.
- FN (False Negative) — это количество объектов, которые модель не смогла обнаружить, хотя они присутствуют на изображении.

По рассмотренным величинам вычисляются следующие метрики:

- $Precision = \frac{TP}{TP+FP}$ – характеризует долю истинно положительных срабатываний (правильно распознанных дорожных знаков) среди всех обнаруженных объектов.
- $Recall = \frac{TP}{TP+FN}$ – измеряет способность модели находить все целевые объекты на изображении.

Для анализа эффективности обученных моделей YOLOv5 и YOLOv8 использовались также следующие метрики:

- Mean Average Precision (mAP) mAP50 и mAP50-95 дают комплексную оценку модели, учитывая как точность детектирования, так и степень перекрытия предсказанного и реального объекта.
  - mAP50 – усредненное значение точности при пороге IOU (Intersection over Union) = 0.5.
  - mAP50-95 – усредненное значение точности при порогах IOU от 0.5 до 0.95 с шагом 0.05.
- Fitness. Комплексная метрика, которая включает в себя взвешенные значения Precision, Recall и mAP. Fitness позволяет объективно сравнивать модели, учитывая их баланс между различными характеристиками.
- Скорость обработки. Для оценки производительности моделей в реальном времени учитывались следующие этапы обработки:
  - Preprocess – время предобработки изображения.
  - Inference – время выполнения детектирования.
  - Postprocess – время постобработки результатов детектирования. Скорость обработки критически важна для систем реального времени, таких как ADAS или автономные транспортные средства.

Эти метрики обеспечивают комплексную оценку качества работы модели, позволяя учитывать как точность и полноту детектирования, так и вычислительную эффективность. Использование нескольких метрик позволяет объективно сравнивать модели и выбирать наиболее подходящую для решения конкретной задачи.

Оценка производительности моделей проводилась с использованием стандартных метрик, таких как Precision, Recall, mAP50, mAP50-95, а также с учетом вычислительной эффективности (время предобработки, инференса и постобработки).

Модель YOLOv5 показала следующие результаты:

- Precision: 0.9668
- Recall: 0.9129
- mAP50: 0.9587
- mAP50-95: 0.7526
- Fitness: 0.7732

Среднее время обработки одного изображения:

- Предобработка: 1.82 мс
- Инференс: 11.01 мс
- Постобработка: 4.32 мс

YOLOv5 продемонстрировала высокую точность при умеренном времени инференса, что делает ее эффективной для приложений, где критичен баланс между точностью и производительностью.

Модель YOLOv8 превзошла YOLOv5 по большинству ключевых метрик:

- Precision: 0.9486
- Recall: 0.9409
- mAP50: 0.9832
- mAP50-95: 0.7971
- Fitness: 0.8157

Среднее время обработки одного изображения:

- Предобработка: 2.05 мс
- Инференс: 10.50 мс
- Постобработка: 6.17 мс

YOLOv8 продемонстрировала улучшение точности и способности обобщать результаты (mAP50-95), что свидетельствует о ее преимуществах в задачах, требующих более высокой точности детектирования.

Таблица 1 отображает количественные оценки для двух подходов.

ТАБЛИЦА 1. Оценка детектирующей части

Метрика	**YOLOv5**	**YOLOv8**
Precision	0.9668	0.9486
Recall	0.9129	0.9409
mAP50	0.9587	0.9832

mAP50-95	0.7526	0.7971
Fitness	0.7732	0.8157
Время предобработки (мс)	1.82	2.05
Время инференса (мс)	11.01	10.50
Время постобработки (мс)	4.32	6.17

YOLOv8 превосходит YOLOv5 по ключевым метрикам качества детектирования, что делает ее более подходящей для задач, требующих высокой точности, таких как системы автономного вождения и ADAS. Однако YOLOv5 демонстрирует более стабильное время обработки, что может быть важным фактором в системах реального времени с ограничениями по вычислительным ресурсам. На рисунках 5 и 6 отражены матрицы ошибок двух моделей.

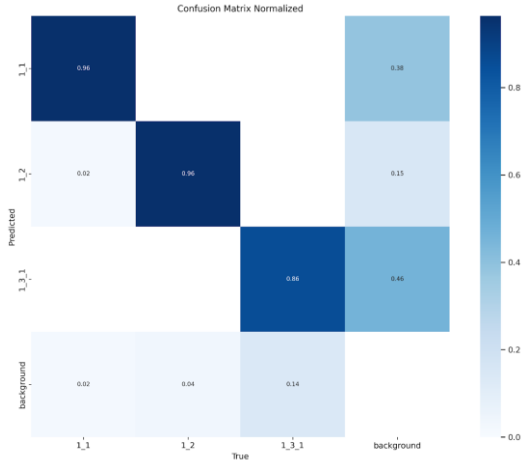


Рис. 5. Матрица ошибок для YOLOv5

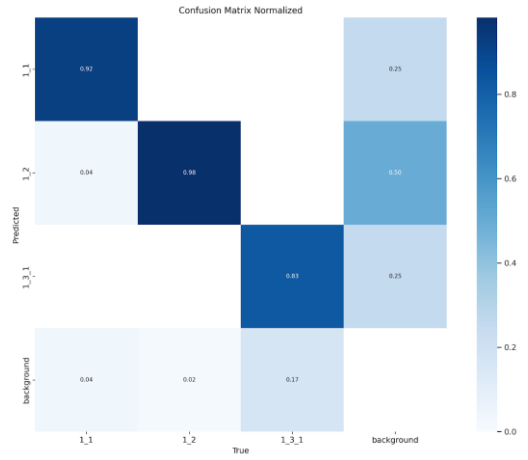


Рис. 6. Матрица ошибок для YOLOv8

V. Заключение

В ходе исследования была проведена оценка возможностей двух версий нейросетевой модели YOLO — YOLOv8 и YOLOv5 — для задачи детектирования дорожных знаков, обозначающих железнодорожные переезды. Модели обучались на специально подготовленном наборе данных с тремя типами знаков, размеченных вручную в системе CVAT. Анализ показал, что:

- YOLOv8 превосходит YOLOv5 по метрикам точности (mAP50 0.9832 против 0.9587) и способности обобщать результаты (mAP50-95: 0.7971 против 0.7526), что делает её более подходящей для задач с высокими требованиями к точности.
- YOLOv5, несмотря на уступающие показатели точности, продемонстрировала более стабильное время обработки, что особенно важно для систем реального времени.

Обе модели показали способность к детектированию знаков в сложных условиях, включая перекрытие объектов, плохую видимость и вариативность масштабов. Эти результаты подтверждают возможность использования современных архитектур YOLO для задач автоматического детектирования дорожных знаков в реальных условиях.

Литература

[1] Карякин, А. В. Исследование возможности классификации дорожных знаков / А. В. Карякин // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 59-61. – EDN FEGRPY.

[2] Никитин, Д. В. Детектирование дорожных знаков на основе нейросетевой модели YOLO / Д. В. Никитин, И. С. Тараненко, А. В. Катаев // Инженерный вестник Дона. – 2023. – № 7(103). – С. 91-99. – EDN MPZLRZ.

[3] Кирвяков, В. О. Исследование возможности детектирования дорожных знаков / В. О. Кирвяков // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 145-150. – EDN YUWXWU.

[4] Али, Б. Алгоритмы навигации беспилотных летательных аппаратов с использованием систем технического зрения / Б. Али, Р. Н. Садеков, В. В. Цодокова // Гироскопия и навигация. – 2022. – Т. 30, № 4(119). – С. 87-105. – DOI 10.17285/0869-7035.00105. – EDN ETCJST.

[5] Использование 3D-сетей для «предсказания» моделей поведения транспортных средств в задаче беспилотного движения трамвая / Н. С. Гужва, В. Е. Прун, В. В. Постников [и др.] // XXIX Санкт-Петербургская международная конференция по интегрированным навигационным системам : сборник материалов, Санкт-Петербург, 30 мая – 01 2022 года. – Санкт-Петербург: "Концерн "Центральный научно-исследовательский институт "Электроприбор", 2022. – С. 304-310. – EDN JQNIIU.

[6] Redmon J. Unified, real-time object detection //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016.

[7] Ren S. Faster r-cnn: Towards real-time object detection with region proposal networks //arXiv preprint arXiv:1506.01497. – 2015.

[8] Z. C. Lipton, C. P.Elkan, B. Narayanaswamy. “Thresholding Classifiers to Maximize F1 Score”, 2014 arXiv: Machine Learning, pp. 1-16.

[9] Jocher G. Yolov5 documentation //docs. ultralytics. com. – 2020. – Т. 5.

[10] Yu F. et al. Bdd100k: A diverse driving video database with scalable annotation tooling //arXiv preprint arXiv:1805.04687. – 2018. – Т. 2. – №. 5. – С. 6.

[11] Behrendt K., Novak L., Botros R. A deep learning approach to traffic lights: Detection, tracking, and classification //2017 IEEE International Conference on Robotics and Automation (ICRA). – IEEE, 2017. – С. 1370-1377.

ГРНТИ 20.53.19
ASJC 1707
OECD 01.02.EP

Распознавание БПЛА различных классов средствами компьютерного зрения

И. М. Бахвалов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2002622@edu.misis.ru

С. В. Старцев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2401133@edu.misis.ru

***Аннотация* — в работе рассматриваются две популярные архитектуры нейронных сетей для задачи распознавания беспилотных летательных аппаратов — БПЛА: YOLOv8L и RT-DETR Large. Для обучения был собран датасет из открытых источников, отражающий различные условия эксплуатации дронов. Проведённый сравнительный анализ показал, что модель RT-DETR Large требует меньше времени на обучение и при этом обеспечивает практически сопоставимые метрики качества (mAP и Recall). Полученные результаты могут послужить основой для выбора оптимальной архитектуры нейронной сети в прикладных задачах, связанных с детекцией и распознаванием БПЛА.**

Ключевые слова — компьютерное зрение, детекция БПЛА, распознавание БПЛА, беспилотные летальные средства, дроны, YOLO.

I. Введение

Изучением и разработкой систем распознавания БПЛА занимаются много научных и исследовательских учреждений, а также ведущие компании в области обороны и технологий по всему миру с конца 1990-х годов.

Одной из ключевых задач при создании систем для мониторинга летательных аппаратов является обнаружение и распознавание объектов на изображения [1]. Для решения этой задачи активно применяются технологии компьютерного зрения. В условиях реального времени важно не только точно обнаружить технику, но и классифицировать её по типу, что требует применения сложных алгоритмов и методов, включая глубокие нейронные сети и другие подходы машинного обучения.

Методы компьютерного зрения предлагают мощное решение для обнаружения беспилотников. Используя передовые алгоритмы обработки изображений и машинного обучения, системы компьютерного зрения могут анализировать изображения и видео, чтобы идентифицировать беспилотники на основе их уникальных визуальных характеристик, таких как форма, размер и характер движения [2].

Решения, основанные на глубоких нейронных сетях, таких как YOLO (You Only Look Once) [3], доказали свою эффективность в задачах компьютерного зрения, включая распознавание объектов в условиях разнообразных внешних факторов. Эти алгоритмы способны быстро и точно анализировать изображения, обеспечивая необходимую информацию для оперативного реагирования, но их исследование и анализ сталкиваются с трудностями, связанными с камуфляжем, маскировкой объектов, а также изменяющимися погодными условиями, которые могут сильно ухудшать качество изображения [4].

На сегодняшний день существуют открытые базы данных, содержащие изображения, однако они ограничены по своему масштабу и разнообразию, что уменьшает возможности для универсализации обученных моделей [5, 6].

II. Наборы данных

Для решения задачи детектирования дронов был собран пользовательский датасет, состоящий из изображений из открытых источников данных. Собранные данные имеют как фотографии дронов в хорошем качестве, так и в плохом, например, снятые кадры низкого качества или при нелетной погоде. Для улучшения качества обучения и повышения разнообразия данных была проведена аугментация, которая включала следующие методы: размытие (Blur), вращение (Rotation) и изменение экспозиции (Exposure).

После применения методов аугментации количество изображений в датасете увеличилось до 1168, что позволило создать более разнообразный и обогащенный набор данных. Для каждой картинки была произведена разметка объектов на 2 класса: дрон самолетного типа (airplane type drone) и дрон вертолетного типа (helicopter type drone).

На рисунках 1-4 изображены примеры рассматриваемых данных в реальных условиях.

Рис. 1. Пример самолетного дрона

Рис. 2. Пример вертолетного дрона

Рис. 3. Пример самолетного дрона при плохой погоде

Рис.4. Пример вертолетного дрона белого цвета на белом фоне

III. Нейросетевые архитектуры

YOLOv8 (You Only Look Once) [7] — это современная модель для распознавания объектов, разработанная компанией Ultralytics. Продолжая традицию серии YOLO, эта версия предлагает улучшенную точность, скорость и эффективность по сравнению с предыдущими выпусками. Основное преимущество YOLOv8 заключается в тщательно продуманной нейронной сети, которая состоит из нескольких ключевых компонентов: Backbone, Neck и Head:

Backbone отвечает за извлечение признаков из входного изображения. В YOLOv8 используется улучшенная архитектура, вдохновлённая CSP (Cross Stage Partial) структурами, которые уже зарекомендовали себя в предыдущих версиях, таких как YOLOv5. Основные особенности Backbone в YOLOv8:

- C2f модули: эти модули сочетают преимущества CSP и FPN (Feature Pyramid Network) [8], что позволяет эффективно передавать информацию между разными уровнями сети и улучшает общую способность модели к обучению.
- Эффективные слои: применяются оптимизированные сверточные слои, например Depthwise Separable Convolutions, которые снижают вычислительную нагрузку без потери точности.
- Активационные функции: используются функции типа SiLU (Sigmoid Linear Unit), которые обеспечивают необходимую нелинейность и способствуют лучшему обучению сети.

Neck служит для объединения признаков, полученных от Backbone, и подготовки их для финальной части сети — Head. В YOLOv8 используется улучшенная версия Path Aggregation Network (PANet), которая обладает следующими характеристиками:

Многоуровневая агрегация: объединение признаков с разных уровней Backbone позволяет модели эффективно обнаруживать объекты разных размеров.

Улучшенная передача информации: новые механизмы соединения слоёв обеспечивают более глубокую и богатую передачу признаков, что повышает точность детекции.

Минимизация потерь: оптимизированные пути передачи признаков помогают сохранить как можно больше информации при агрегации.

Head отвечает за окончательное предсказание местоположения объектов, их классов и вероятностей. В YOLOv8 реализованы следующие элементы:

- Anchor-Free подход: в отличие от предыдущих версий, YOLOv8 использует подход без якорей, что упрощает процесс обучения и снижает количество гиперпараметров.
- Прогнозирование ограничивающих рамок: сеть напрямую предсказывает координаты bounding boxes с помощью регрессии.
- Классификационные слои: определяются вероятности принадлежности объектов к различным классам.
- Использование NMS (Non-Maximum Suppression): для удаления дублирующихся предсказаний и выбора наиболее вероятных объектов применяется улучшенный алгоритм NMS.

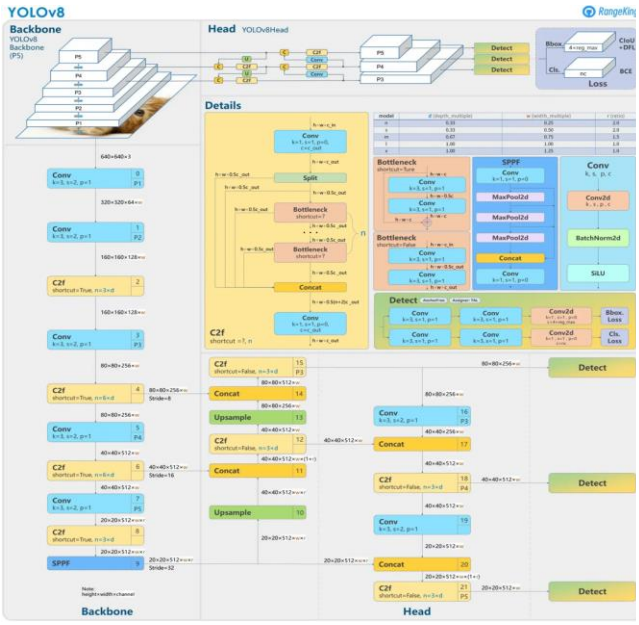


Рис.5. Архитектура YOLOv8

Сначала происходит извлечение признаков. Backbone YOLOv8 использует серию сверточных слоёв с остаточными связями и CSP-модулями. Это позволяет эффективно извлекать многоуровневые признаки из изображения, сохраняя при этом высокую вычислительную эффективность. Использование Depthwise Separable Convolutions помогает снизить количество параметров и ускорить вычисления без значительной потери точности.

Следующим этапом происходит агрегация признаков Neck сети, которая объединяет признаки, полученные из различных слоев Backbone, используя PANet. Это обеспечивает более глубокую интеграцию информации и улучшает способность модели обнаруживать объекты разных размеров. Дополнительно применяются механизмы Feature Pyramid Networks (FPN) для создания пирамиды признаков, которая помогает модели лучше справляться с вариативностью объектов.

Последним шагом является предсказание объектов. Head сети выполняют конечное предсказание объектов. В YOLOv8 используется anchor-free метод, который предсказывает координаты bounding boxes напрямую, без необходимости подбора якорей (anchors). Это упрощает процесс обучения и уменьшает количество гиперпараметров, что делает модель более гибкой и устойчивой к различным типам данных. YOLOv8 — это первая модель без якорей.

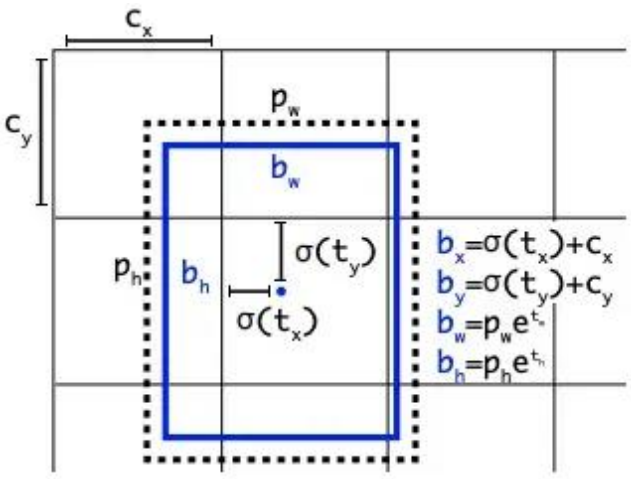


Рис.6. Визуализация якорного бокса в YOLO

Якорные боксы были известной проблемной частью предыдущих моделей YOLO, поскольку они могли представлять распределение боксов целевого эталона, но не распределение пользовательского набора данных.

В каждой категории моделей YOLOv8 есть пять моделей для обнаружения, сегментации и классификации. YOLOv8 Nano - самый быстрый и маленький, в то время как YOLOv8 Extra Large (YOLOv8x) - самый точный, но самый медленный среди них.

YOLOv8 поставляется в комплекте со следующими предварительно подготовленными моделями:

- Контрольные точки обнаружения объектов обучены на основе набора данных COCO detection с разрешением изображения 640.
- Контрольные точки сегментации экземпляра, обученные на наборе данных сегментации COCO с разрешением изображения 640.
- Модели классификации изображений предварительно обучены на базе данных ImageNet с разрешением изображения 224.

Функции потерь (loss functions) играют ключевую роль в обучении нейронных сетей, определяя, насколько хорошо модель предсказывает целевые значения. В YOLOv8 используются адаптивные функции потерь, которые специально разработаны для улучшения точности и эффективности модели при решении задачи обнаружения объектов. Рассмотрим подробнее основные функции потерь, применяемые в YOLOv8:

1. CIoU (Complete Intersection over Union)

CIoU — это расширенная версия метрики IoU (Intersection over Union), которая учитывает не только перекрытие между предсказанным и истинным ограничивающими рамками (bounding boxes), но и геометрические аспекты их расположения [9].

Перекрытие (Intersection over Union) — стандартная метрика IoU измеряет степень перекрытия между предсказанным и истинным боксом. Чем выше IoU, тем лучше совпадение.

Расстояние между центрами: CIoU учитывает расстояние между центрами предсказанного и

истинного боксов. Это помогает модели лучше позиционировать объекты, особенно когда боксы не перекрываются полностью.

Соотношение сторон — CIoU учитывает разницу в пропорциях (ширина и высота) между предсказанным и истинным боксами. Это способствует более точному соответствию формы объектов.

$$\text{CIoU} = \text{IoU} - \frac{\rho^2!\,(b, b^{\text{gt}})}{c^2} - \alpha\, v$$

где:

- b, b^{gt} — Евклидово расстояние между центрами предсказанного и истинного боксов.
- c — диагональ наименьшего ограничивающего прямоугольника, который охватывает оба бокса.
- v — мера различия в аспектах боксов.
- α — весовой коэффициент, зависящий от v

Преимущества использования CIoU:

- Гибкость настройки :параметры α и γ позволяют адаптировать функцию потерь под конкретные задачи и распределение данных. Более точное позиционирование: учет расстояния между центрами и соотношения сторон позволяет модели лучше локализовать объекты.
- Стабильное обучение: CIoU обеспечивает более гладкую градиентную поверхность, что способствует стабильности процесса обучения
- Улучшенная сходимость: быстрая и точная адаптация боксов к объектам за счет дополнительных геометрических компонентов.

2. Focal Loss

Focal Loss — это модификация функции потерь для классификации, разработанная для решения проблемы несбалансированности классов, особенно актуальной в задачах обнаружения объектов, где количество фона (negative samples) значительно превышает количество объектов (positive samples) [10].

Focal Loss обладает фокусировкой на сложных примерах — уменьшает вклад легко классифицируемых примеров (с высоким значением вероятности правильного класса) и фокусируется на трудных для классификации примерах, а так же обладает адаптивным снижением веса - введение фактора, который снижает вес легко классифицируемых примеров, что позволяет модели лучше обучаться на сложных и неправильно классифицированных примерах.

Формула Focal Loss:

$$\text{FL}(p_t) = -\alpha_t\,(1 - p_t)^{\gamma}\,\log(p_t)$$

где:

- p(t) — вероятность правильного класса.
- α(t) — весовой коэффициент для класса t, позволяющий балансировать важность различных классов.
- γ — параметр, контролирующий степень фокусировки. При γ=0 Focal Loss сводится к стандартной кросс-энтропии.

Преимущества использования Focal Loss:

- Снижение влияния фона: путем уменьшения веса легко классифицируемых примеров фоновых классов, Focal Loss позволяет модели уделять больше внимания объектным классам.
- Улучшенная производительность: фокусировка на сложных примерах способствует лучшему обучению и повышению точности классификации.
- Гибкость настройки: параметры α и γ позволяют адаптировать функцию потерь под конкретные задачи и распределение данных.

3. Дополнительные функции потерь

Помимо CIoU и Focal Loss, в YOLOv8 могут использоваться и другие функции потерь для различных аспектов задачи обнаружения объектов:

Классификационная потеря (Classification Loss): обычно основана на Focal Loss или кросс-энтропии для предсказания классов объектов.

Регрессионная потеря (Regression Loss): помимо CIoU, могут применяться функции, учитывающие расстояние и размер боксов, такие как GIoU или DIoU.

Потери за объектность (Objectness Loss): оценивают вероятность наличия объекта в определенных боксах, помогая модели отличать объекты от фона.

Архитектура RT-DET

Real-Time Detection Transformer (RT-DETR), разработанный компанией Baidu, представляет собой передовой комплексный детектор объектов, обеспечивающий производительность в реальном времени при сохранении высокой точности. Он основан на идее DETR (фреймворк без NMS), но при этом в него введена основа на conv и эффективный гибридный кодер для достижения скорости в реальном времени.

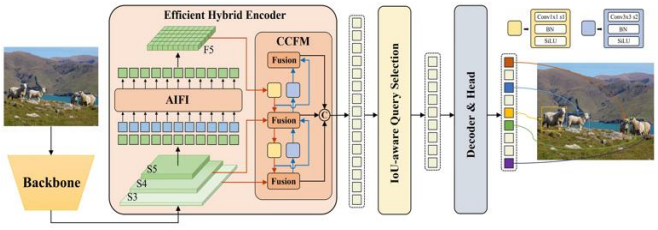


Рис.7. Визуализация якорного бокса в YOLO

На схеме архитектуры модели RT-DETR показаны три последних этапа магистрали {S3, S4, S5} в качестве входа для кодера. Эффективный гибридный кодер преобразует разномасштабные признаки в последовательность признаков изображения с помощью внутримасштабного взаимодействия признаков (AIFI) и модуля межмасштабного слияния признаков (CCFM). Выбор запроса с учетом IoU используется для выбора фиксированного числа признаков изображения, которые служат исходными объектными запросами для декодера. Наконец, декодер со вспомогательными головками предсказания итеративно оптимизирует

запросы к объектам для получения боксов и оценок уверенности.

IV. Сравнение

Для решения задачи обнаружения объектов были выбраны две нейросетевые архитектуры: YOLO (You Only Look Once) и RE-DETR (Real-Time Detection Transformer). Эти подходы были выбраны благодаря их высокой эффективности в задачах компьютерного зрения и способности быстро и точно идентифицировать объекты в сложных сценах. YOLO известен своей скоростью и точностью, что делает его идеальным для реального времени, в то время как RE-DETR является трансформером для улучшения обработки контекста и взаимодействия между объектами, что позволяет достигать более высоких результатов в сложных условиях.

Для оценки качества распознавания объектов обычно используются три ключевые метрики. Первая из них — точность (Precision), которая показывает долю правильно предсказанных объектов среди всех предсказаний. Вторая метрика — полнота (Recall), отражающая долю верно обнаруженных объектов среди всех существующих объектов. Наконец, mAP (mean Average Precision) представляет собой среднее значение Average Precision для каждого класса, что можно интерпретировать как площадь под кривой Precision-Recall. Эти метрики позволяют комплексно оценить эффективность моделей распознавания объектов.

$$\text{Precision} = \frac{\text{TP}(c)}{\text{TP}(c) + \text{FP}(c)}$$

$$\text{Recall} = \frac{\text{TP}(c)}{\text{TP}(c) + \text{FN}(c)}$$

где $TP(c)$ - количество предсказаний True Positive для класса c, $FP(c)$ - количество предсказаний False Positive для класса c, $FN(c)$ - количество предсказаний False Negative для класса c.

Были обучены 2 модели YOLOv8L и RT-DETR Large с помощью фреймоворка Ultralystics. Обучение производилось на изображениях размера 640x640 и 100 эпохах для первой модели и 45 эпох для второй модели. В процессе обучения использовалась аугментация для достижения более лучших результатов.

Полученные метрики в результате обучения приведены на рисунке 6. В качестве порога IoU использовалось значение 0.7, а в качестве порога Confiednce использовалось значение 0.4 для Precision Recall.

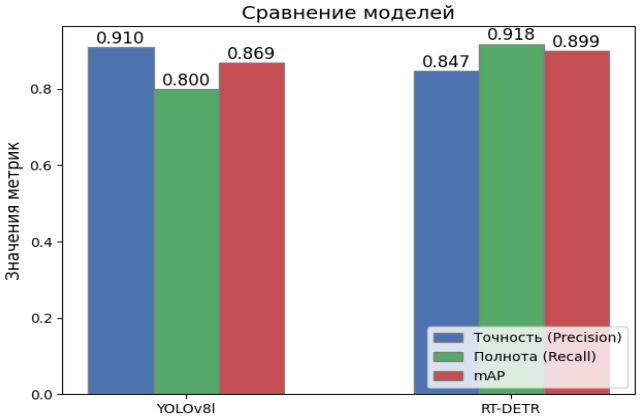


Рис.8. сравнение моделей по метрикам

В ходе обучения модели YOLOv8L было установлено, что она демонстрирует в среднем более высокую точность (Precision) по сравнению с моделью RT-DETR. Однако стоит отметить, что YOLOv8L уступает RT-DETR по метрике Recall. Значения метрики mAP (mean Average Precision) для обеих моделей оказываются практически идентичными.

На рисунке 9 показаны графики, иллюстрирующие зависимость метрик и ошибок от числа эпох обучения, построенные на тестовом наборе данных для модели YOLOv8L. Из представленных графиков видно, что значения метрик продолжают увеличиваться, в то время как ошибки остаются на стабильном уровне, что указывает на отсутствие переобучения модели.

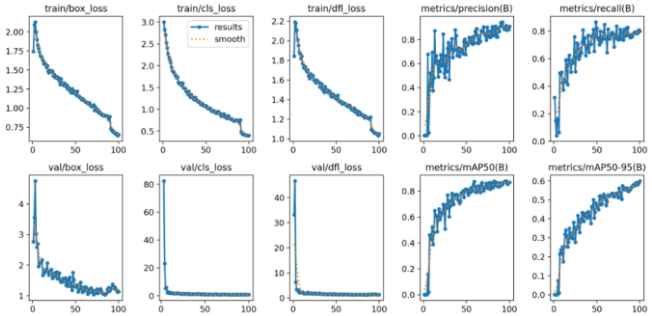


Рис. 9. Зависимость значений метрик и ошибок от эпох в процессе обучения модели YOLOv8L

На рисунке 10 показан такой же график, как и на рисунке 9, но для модели RT-DETR. Из представленных графиков видно, что значения метрик продолжают увеличиваться, в то время как ошибки остаются на стабильном уровне, что указывает на отсутствие переобучения модели.

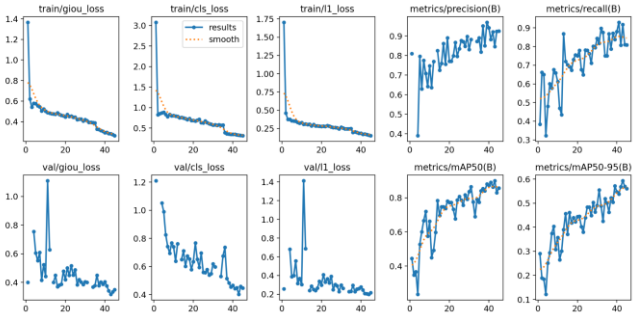


Рис. 10. Зависимость значений метрик и ошибок от эпох в процессе обучения модели RT-DETR Large.

Более подробная статистика по двум моделям представлена в нормализованных матрицах ошибок, показанных на рисунке 11 и 12. В рамках задачи распознавания объектов данная матрица иллюстрирует соотношение между предсказаниями модели и истинными метками классов.

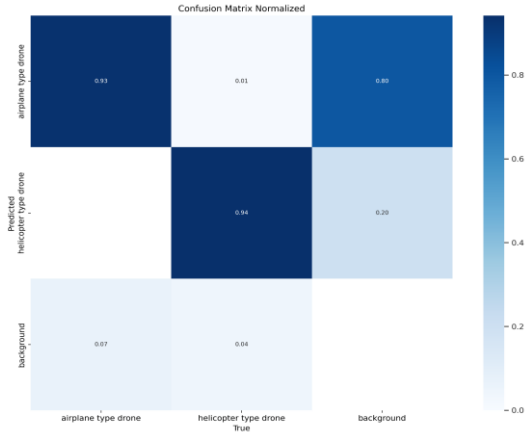


Рис.11. Нормализованная матрица ошибок RT-DETR

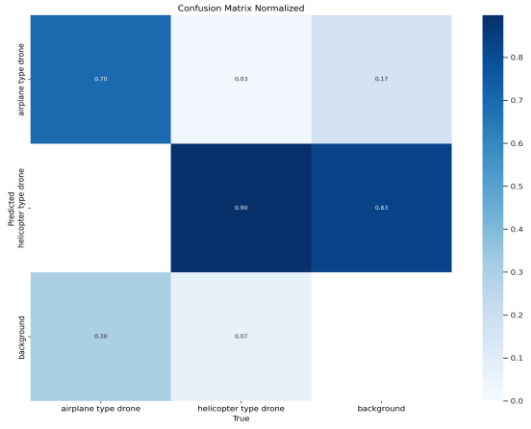


Рис.12. Нормализованная матрица ошибок YOLOv8L

Результат работы дообученной нейросети YOLOv8L представлен на рисунках 13 – 15:

Рис.13. Результат работы дообученной нейросети YOLOv8L

Рис. 14. Результат работы дообученной нейросети YOLOv8L

Рис. 15. Результат работы дообученной нейросети YOLOv8L

Примеры результатов работы дообученной нейросети RT-DETR представлен на рисунках 16 - 18:

Рис. 16. Результат работы дообученной нейросети RT-DETR

Рис. 17. Результат работы дообученной нейросети RT-DETR

Рис. 18. Результат работы дообученной нейросети RT-DETR

V. Заключение

В рамках данного исследования было проведено сравнение двух популярных архитектур нейронных сетей — YOLOv8L и RT-DETR Large — в контексте задачи распознавания объектов.

Датасет для экспериментов был собран из открытых источников и включает разнообразные изображения объектов в различных условиях.

Для оценки качества моделей были использованы такие метрики: precision, recall и mAP. Анализ результатов показал, что модель RT-DETR Large имеет небольшое преимущество почти по всем рассчитанным метрикам. Однако, стоит отметить, что модель YOLOv8L продемонстрировала более высокие значения precision.

Однако при анализе времени обучения моделей было обнаружено существенное различие: обучение модели YOLOv8L заняло 100 эпох, в то время как модель RT-DETR Large была обучена за 45 эпох. Учитывая незначительную разницу в метриках качества и существенную экономию времени при обучении, можно сделать вывод, что для задач распознавания дронов предпочтительнее использовать модель RT-DETR Large. Это позволяет достичь практически идентичного уровня точности с существенным сокращением времени на обучение модели. Полученные результаты могут быть полезны при выборе оптимальной модели для решения аналогичных прикладных задач.

Литература

[1] Ali, B., Sadekov, R.N. & Tsodokova, V.V. A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems. Gyroscopy Navig. 13, 241–252 (2022). https://doi.org/10.1134/S2075108722040022

[2] Pazychev, Dmitry & Bakulev, K. & Sadekov, Rinat. (2023). Low-Cost Navigation System for UAV. 1-6. 10.23919/ICINS51816.2023.10168469

[3] Ultralytics. *YOLOv8 Official GitHub Repository*. https://github.com/ultralytics/ultralytics

[4] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-5, doi: 10.23919/ICINS51816.2023.10168407.F

[5] https://www.researchgate.net/publication/357723173_Detection_and_Recognition_of_Drones_Based_on_a_Deep_Convolutional_Neural_Network_Using_Visible_Imagery

[6] https://www.researchgate.net/publication/364766917_Drone_Detection_and_Tracking_in_Real-Time_by_Fusion_of_Different_Sensing_Modalitis

[7] Terven, Juan & Cordova-Esparza, Diana-Margarita & Romero González, Julio. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 5. 1680-1716. 10.3390/make5040083.

[8] T.-Y. Lin, P. Dollár, R. Girshick, et al. *Feature Pyramid Networks for Object Detection.* CVPR, 2017.

[9] Z. Zheng, P. Wang, W. Liu, et al. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. AAAI, 2020

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár. Focal Loss for Dense Object Detection. IEEE TPAMI, 2020

ГРНТИ 28.23.15
ASJC 1707
OECD 01.02.EP

Особенности детектирования знаков дорожного движения «Пешеходный переход»

С. С. Белякова
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2414121@edu.misis.ru

***Аннотация* — данная работа посвящена актуальной проблеме разработки эффективных систем распознавания дорожных объектов, в частности, знаков пешеходного перехода, для беспилотных транспортных средств. В статье исследуется применение сверточной нейронной сети YOLOv8, зарекомендовавшей себя в задачах детекции объектов, для решения поставленной задачи. Исследование направлено на оценку применимости и потенциала YOLOv8 для обеспечения надежной и точной детекции знаков пешеходного перехода в реальных условиях.**

Ключевые слова — компьютерное зрение, детекция дорожных знаков, распознавание знаков пешеходного перехода, YOLO.

I. Введение

Развитие беспилотных технологий управления транспортом ставит перед исследователями и инженерами ряд сложных технических задач. Одним из важнейших вызовов является создание систем восприятия окружающей среды[1], способных надежно и точно распознавать различные объекты[2,3], включая элементы дорожной инфраструктуры. Беспилотные транспортные средства должны уметь распознавать как другие автомобили и пешеходов, так и дорожные знаки, например, указатели пешеходных переходов. Это особенно важно для обеспечения безопасного и эффективного движения в городской среде.

Современные методы машинного обучения, в частности, глубокое обучение, показывают значительные успехи в решении подобных задач. Сверточные нейронные сети (CNN), архитектура которых позволяет эффективно обрабатывать изображения, стали стандартом для многих задач компьютерного зрения. Среди доступных архитектур CNN особое внимание заслуживает YOLO (You Only Look Once), которая позволяет достигать высокой скорости и точности детекции объектов в реальном времени.

В данной работе исследуется применение одной из последних версий архитектуры YOLO, а именно YOLOv8, для решения задачи распознавания знаков пешеходного перехода. Данный выбор обусловлен ее эффективностью и способностью обрабатывать изображения в реальном времени, что является ключевым требованием для беспилотных транспортных средств.

II. YOLOv8

Для обучения на рассмотренном датасете была выбрана модель YOLOv8[4]. YOLOv8 является версией модели YOLO, разработанной компанией Ultralytics. Эта передовая модель, относящаяся к категории state-of-the-art (SOTA), основана на достижениях предыдущих версий и включает в себя новые функции и улучшения, направленные на повышение производительности, гибкости и эффективности. YOLOv8 охватывает широкий спектр задач в области искусственного интеллекта, включая детекцию объектов, сегментацию, оценку позы, отслеживание и классификацию.

Архитектура YOLOv8 основана на сверточной нейронной сети, которая делится на два основных компонента: позвоночник (backbone) и голову (head). Позвоночник представляет собой модифицированную версию архитектуры CSPDarknet53[5], состоящую из 53 сверточных слоев, использующих частичные межэтапные соединения для улучшения информационного потока между слоями. Голова YOLOv8 включает несколько сверточных слоев, за которыми следуют полносвязные слои, отвечающие за прогнозирование ограничивающих прямоугольников, оценку вероятности классов для обнаруженных объектов на изображении. Одной из ключевых особенностей YOLOv8 является механизм самоконтроля в голове сети, который позволяет модели акцентировать внимание на различных частях изображения и определять значимость признаков в зависимости от их актуальности для задачи. Кроме того, YOLOv8 обладает способностью выполнять многомасштабное обнаружение объектов, используя пирамидальную сеть признаков, что позволяет эффективно обнаруживать объекты различных размеров и масштабов на изображении.

На рисунке 1 представлена детальная визуализация архитектуры нейронной сети YOLOv8.

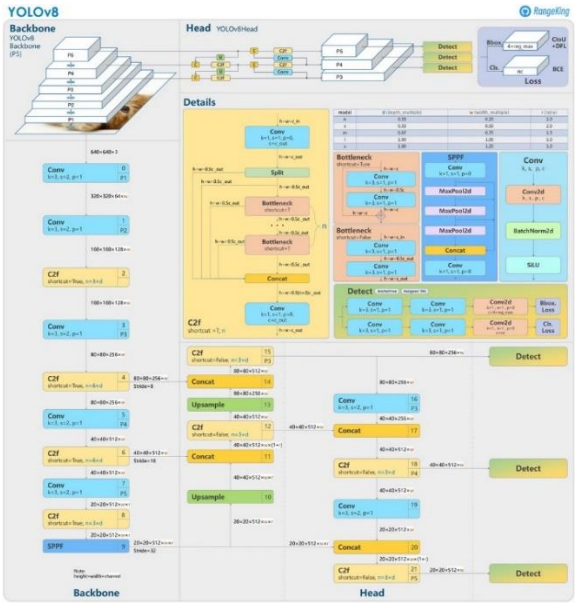


Рис. 1. Архитектура нейронной сети YOLOv8

III. Наборы данных

Для обучения рассмотренной в данной работе нейросети использовались открытый набор данных и собранный самостоятельно. Рассмотрим используемые наборы данных:

A. RTS

RTS[6] (Russian traffic signs) – набор данных, предназначенный для обучения моделей для распознавания дорожных знаков, включая знаки пешеходного перехода и другие встречающиеся на территории РФ важные дорожные указатели.

Датасет содержит 2390 изображений и имеет 156 классов знаков, встречающихся на изображении. Набор данных содержит фотографии, снятые при разном освещении и погодных условиях. Примеры таких фотографий представлены на рисунке 2.

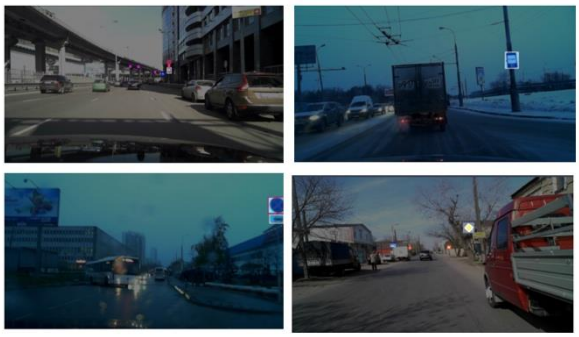


Рис. 2. Примеры изображений из набора данных.

В рамках данной работы целевыми объектами являются знаки пешеходного перехода, их виды представлены на рисунке 3. В наборе данных присутствует 454 изображения, содержащих данные знаки, более половины из которых имеют на себе оба подвида знаков пешеходного перехода.

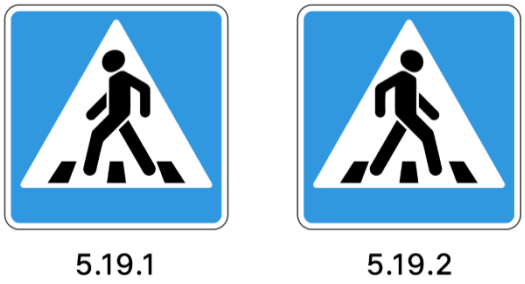


Рис. 3. Виды знаков пешеходного перехода

Б. Личный набор данных

Данный датасет состоит из 205 изображений, на каждом из которых присутствуют знаки пешеходного перехода. Изображения были получены из набора видео, снятых при движении трамвая. Видео были нарезаны на кадры, и из них уже составлялся датасет. После отсева изображения были размечены при помощи онлайн-ресурса CVAT [7], объединены вместе с предыдущим датасетом и разделены на train, validation и test в соотношении 65%, 20% и 15%. На рисунке 4 представлены примеры размеченных изображений из данного датасета.

Рис. 4. Примеры изображений из личного набора данных.

Данный набор аналогично предыдущему имеет изображения с разными погодными условиями и освещенностью.

IV. Обучение и результаты

Из моделей YOLOv8 была взята YOLOv8n (Nano) и обучена на полученном датасете на 150 эпохах в течение 70 минут. Рассмотрим некоторые метрики [8] полученной модели. На рисунке 5 представлены графики изменения значения функции потерь по эпохам. По данным графикам видно, что значение функции потерь уменьшается в течение обучения, обозначая этим эффективное обучение модели. Из графиков видно, что примерно к 140 эпохе начинается переобучение модели (значение функции потерь начинает увеличиваться).

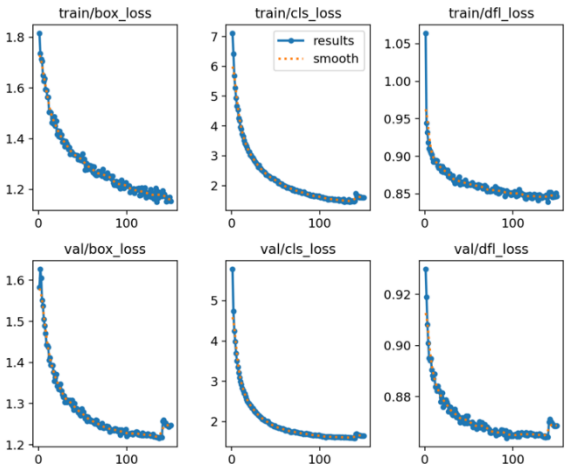


Рис. 5. Кривые функции потерь, сверху обучающая выборка, снизу валидационная.

Для оценки качества работы модели была выбрана F1-мера. F1-мера – среднее гармоническое от precision и recall, обеспечивающее сбалансированную оценку эффективности модели с учетом как ложноположительных, так и ложноотрицательных результатов. Рассмотрим ее и ее составляющие подробнее:

- TP (True positive) – количество истинно положительных случаев (когда объект был верно обнаружен и принадлежит целевому множеству)
- FP (False positive) – количество ложноположительных случаев (обнаружение нецелевого объекта как целевого)
- FN (False negative) – количество ложноотрицательных случаев (классификация элемента целевого множества как нецелевого)
- TN (True negative) – количество истинно отрицательных случаев (объект верно классифицирован как не принадлежащий целевому множеству)
- Precision (Точность) – определяет долю истинно положительных прогнозов среди всех положительных прогнозов, оценивая способность модели избегать ложных срабатываний.
  P = TP/(TP+FP)
- Recall (Полнота) – рассчитывает долю истинных положительных результатов среди всех реальных положительных результатов, оценивая способность модели обнаруживать все экземпляры класса.
  R = TP/ (TP+FN)
- F1-мера – баланс между двумя предыдущими метриками:

$$F_\beta = (1+\beta^2)\cdot\frac{precision\cdot recall}{(\beta^2\cdot precision)+recall}$$

Таблица 1 – метрики полученной модели

Знак	P	R	F1
Класс 1	0.781	0.745	0.763
Класс 2	0.712	0.768	0.739

V. Заключение

В данной статье была рассмотрена важность и особенности детектирования российских знаков пешеходного перехода в контексте современных технологий компьютерного зрения. Обнаружение и распознавание этих знаков играют ключевую роль в обеспечении безопасности дорожного движения, особенно в условиях, когда автономные транспортные средства становятся все более распространенными.

Была проанализирована одна из моделей глубокого обучения - YOLOv8, которая демонстрирует высокую точность и эффективность в задачах распознавания объектов. Кроме того, был создан датасет, включающий в себя как открытый набор данных, так и локально сделанный.

В результате обучения модели было получено, что модель хорошо справляется с определением знака пешеходного перехода, но не всегда корректно определяет его вид. Также в открытом датасете присутствует множество других определяемых знаков из-за чего скорость обучения модели на таком наборе данных снижается.

Можно сделать вывод , что детектирование знаков пешеходного перехода является важной задачей, требующей комплексного подхода и использования передовых технологий. Успешная реализация таких систем не только повысит безопасность на дорогах, но и станет важным шагом к развитию автономных транспортных средств и умных городов.

Литература

1. Использование систем технического зрения для определения положения транспортного средства на дороге / Н. И. Котов, С. Б. Беркович, Р. Н. Садеков [и др.] // XXIV Санкт-Петербургская международная конференция по интегрированным навигационным системам : Сборник материалов, Санкт-Петербург, 29–31 мая 2017 года / Главный редактор В.Г. Пешехонов. – Санкт-Петербург: "Концерн "Центральный научно-исследовательский институт "Электроприбор", 2017. – С. 24-27.
2. Лим, В. Л. Исследование вопроса распознавания светофоров / В. Л. Лим // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях: СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 34-39.
3. Кирвяков В.О. Исследование возможности детектирования дорожных знаков // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях: СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 145-150.

4. YOLOv8 - Ultralytics YOLO Docs // Ultralytics YOLOv8 URL: https://docs.ultralytics.com/ru/models/yolov8/ (дата обращения: 10.10.2024).
5. Bochkovskiy A., Chien-Yao W., Hong-Yuan M. L. YOLOv4: Optimal Speed and Accuracy of Object Detection // 2020
6. russian-traffic-signs-recognition // Roboflow URL: https://universe.roboflow.com/mguogareva/russian-traffic-signs-recogntion/browse?queryText=&pageSize=50& startingIndex=850&browseQuery=true (дата обращения: 15.11.2024)
7. Open Data Annotation Platform // CVAT URL: https://www.cvat.ai/ (дата обращения: 20.11.2024).
8. Hossin M., Sulaiman M.N. A Review on Evaluation Metrics for Data Classification Evaluations // International Journal of Data Mining & Knowledge Management Process. - 2015. - №5

ГРНТИ 28.23.37
ASJC 1707
OECD 10201

Применение компьютерного зрения для детекции загрязненных зон пляжей

А.О. Васильева
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2009903@edu.misis.ru

М. Гримм
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2007014@edu.misis.ru

***Аннотация* — загрязнение пляжей создаёт серьёзные экологические и экономические проблемы. Оно наносит ущерб морской экосистеме, снижает привлекательность пляжей и угрожает здоровью людей. В данной работе рассматривается применение методов компьютерного зрения для автоматизации очистки пляжей. Проведён сравнительный анализ моделей YOLOv8 и Faster R-CNN с использованием открытого датасета TACO, содержащего аннотированные изображения мусора. Оценка проводилась по метрикам mAP, Precision и Recall. YOLOv8 демонстрирует высокую скорость работы, что делает её подходящей для задач реального времени. Faster R-CNN обеспечивает более точную детализацию объектов. В статье представлены рекомендации по выбору модели для различных приложений.**

Ключевые слова — компьютерное зрение, YOLOv8, Faster R-CNN, идентификация мусора, TACO, мониторинг пляжей.

I. Введение

Экологическая ответственность и снижение объёмов отходов стали ключевыми направлениями в стратегии многих крупных компаний. Эти инициативы находят поддержку среди покупателей и укрепляют репутацию брендов. Компании, такие как Apple, Coca-Cola, Unilever, Procter & Gamble, Walmart и IKEA, активно внедряют политику переработки отходов и экологизации производства.

Одним из важных шагов в этом направлении является автоматизация классификации и сортировки отходов. Для этого используются технологии компьютерного зрения, которые позволяют эффективно распознавать физические свойства объектов, такие как материал, форма, цвет и прозрачность. Эти параметры используются для отнесения отходов к определённой категории и их дальнейшей переработки [1].

Методы глубокого обучения показали высокую эффективность в задачах классификации и обнаружения объектов. Такие подходы позволяют создавать детекторы для различных сфер, включая дорожное движение, железнодорожный транспорт, авиацию, медицину, биологию и городскую инфраструктуру [2]. Среди них особое внимание привлекают разработки для автоматической классификации мусора по 2D-изображениям, которые можно адаптировать для реальных условий [3, 4, 5].

Однако алгоритмы, основанные на глубоких нейронных сетях, требуют значительных объёмов аннотированных данных и вычислительных ресурсов. Несмотря на доступность зарубежных датасетов для классификации отходов, задача их адаптации и использования в локальных условиях остаётся актуальной. Настоящая работа направлена на изучение одного из таких подходов с целью оценки его применения для автоматизированной сортировки отходов в реальных условиях.

II. Наборы данных

Для исследования использовался расширенный датасет TACO (Trash Annotations in Context), который изначально содержал 1500 изображений мусора, собранных в условиях реального мира, таких как пляжи, леса и дороги [6]. Изображения в оригинальной версии датасета аннотированы по 60 категориям, которые дополнительно сгруппированы в 28 суперкатегорий. Размеры изображений варьируются от 842×474 до 6000×4000 пикселей, что делает данные разнообразными.

Благодаря усилиям сообщества, в данной статье использовался увеличенный датасет с добавленными изображениями, что позволило расширить объем данных и сделать модель более устойчивой.

В наборе данных представлены изображения с мусором, находящимся на различных поверхностях: песке, траве, тротуаре или внутри жилых помещений. Это разнообразие данных позволяет модели лучше адаптироваться к реальным условиям и повышает точность её работы при детекции объектов. Пример разметки изображений датасета представлен на рис. 1.

Для данной задачи было решено сфокусироваться на 18 самых крупных категориях, так как распределение аннотаций по классам оказалось неравномерным. Это позволило упростить задачу, уменьшив влияние малочисленных классов, и сосредоточиться на улучшении качества детекции для наиболее представленных категорий [7]. Распределение аннотаций по этим классам можно увидеть на рис. 4.

Ключевые характеристики датасета TACO:

- разрешение изображений варьируется от 842×474 до 6000×4000 пикселей;
- изображения были собраны в различных природных и городских условиях;

- датасет включает категории, такие как «пластиковые пакеты», «бутылки», «металлические банки» и другие, что актуально для задач экологического анализа;
- в процессе обучения использовались методы увеличения данных (вращение, масштабирование, добавление шумов), что позволило повысить эффективность модели и справиться с дисбалансом классов [8].

Рис. 1. Примеры кадров и их разметки в датасете TACO

Помимо различных поверхностей, этот набор данных включает изображения, сделанные в условиях различного освещения, от яркого дневного света до слабого искусственного освещения.

Рис. 2. Примеры данных при разном освещении в датасете TACO

Кроме того, мусор представлен различных размеров — от мелких объектов, таких как крышки от бутылок, до крупных, например, картонные коробки.

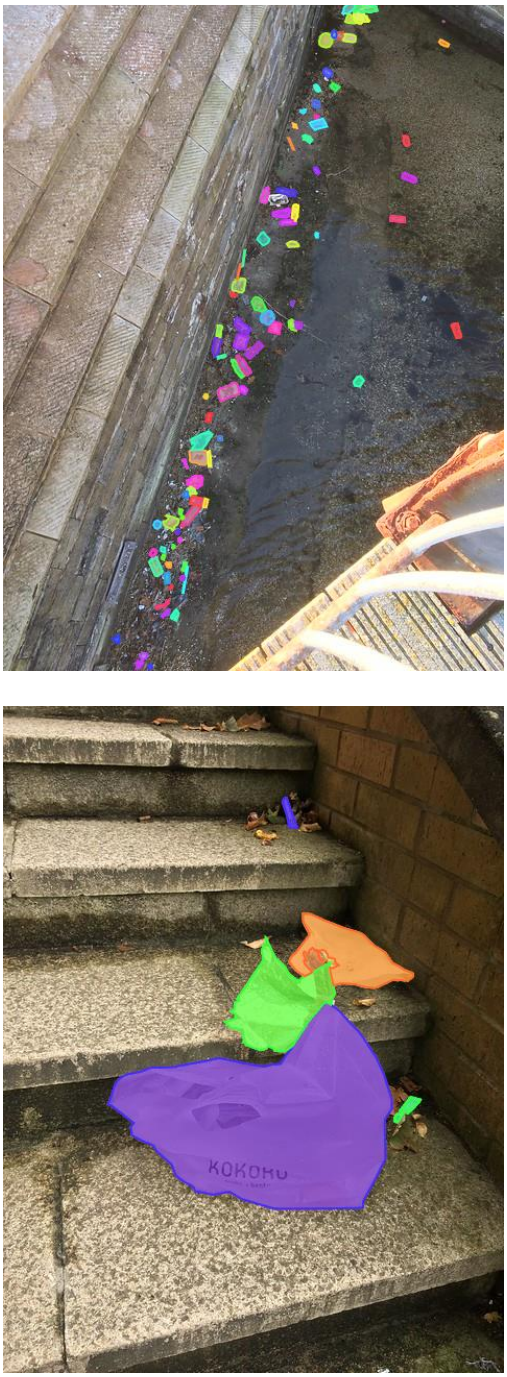


Рис. 3. Примеры данных при различном размере объектов в датасете TACO

Для обучения модели были выделены следующие 18 ключевых классов: алюминиевая фольга, крышка от бутылки, бутылка, разбитое стекло, банка, картон, сигарета, чашка, крышка, другой мусор, другой пластик, бумага, пластиковый пакет или обертка, пластиковый контейнер, язычок от банки, соломинка, кусок пенопласта и неразмеченный мусор.

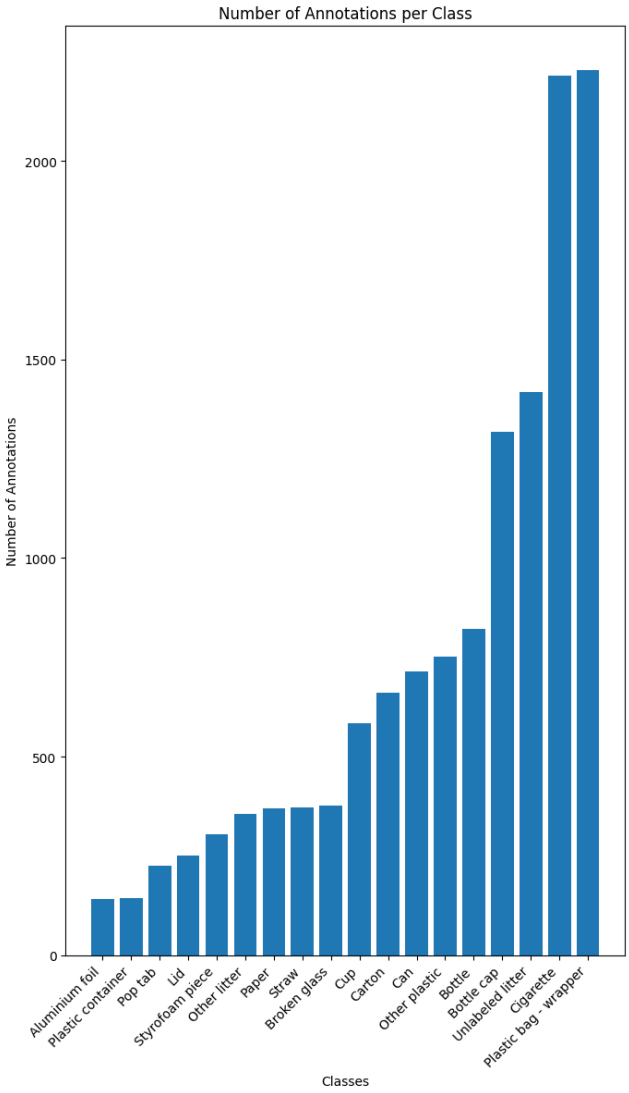


Рис. 4. Распределение изображений по классам в датасете TACO

III. Нейросетевые архитектуры

Для сравнения были обучены две модели: YOLOv8 и FasterCNN.

A. YOLOv8

YOLOv8 (You Only Look Once) представляет собой последнюю версию серии детекторов объектов в реальном времени, демонстрирующую высокую точность и скорость работы. Опираясь на успехи предыдущих поколений YOLO, эта версия включает усовершенствования и новые функции, которые делают её универсальным инструментом для решения задач обнаружения объектов в самых разных областях применения [9].

YOLOv8 поддерживает различные варианты модели, такие как nano, small, medium, large, что позволяет использовать её как на устройствах с ограниченными ресурсами, так и в мощных системах. Для данного исследования была выбрана YOLOv8s благодаря ее сбалансированной архитектуре, которая сочетает в себе высокую производительность и компактность.

Network	Size (Pixels)	aMP^{val} (50–95)	Speed CPU (ms)	Speed T4 GPU (ms)	Params (M)	Flop$_s$ (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

Рис. 5. Сравнение различных версий YOLOv8

Архитектура YOLOv8 разделена на несколько частей:

- Backbone: извлекает глубокие пространственные признаки из входного изображения.
- Neck: интегрирует признаки разных уровней, чтобы создать богатые контекстные признаки.
- Head: делает прогнозы по обнаруженным объектам (классификация, локализация, уверенность).

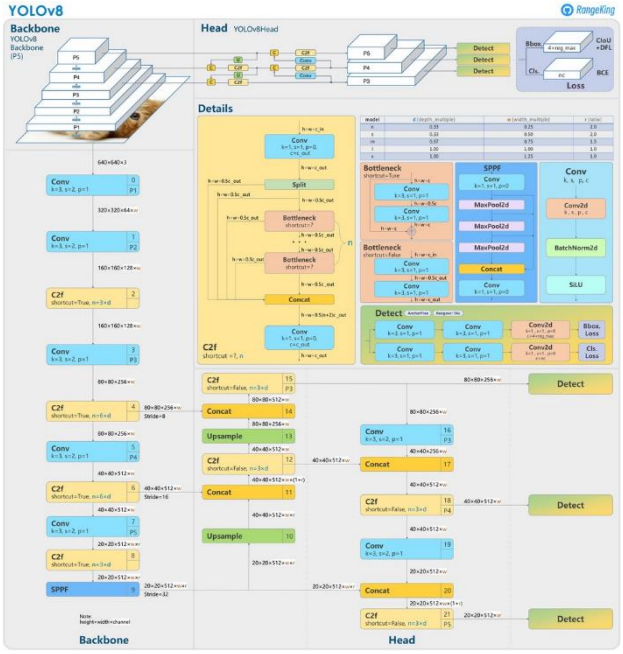


Рис. 6. Архитектура YOLOv8

YOLOv8 использует пирамиду признаков (рис. 6) для повышения точности обнаружения объектов разных размеров. Эта архитектурная концепция позволяет модели анализировать изображения на различных масштабах, улучшая распознавание как крупных, так и мелких объектов.

Пирамида признаков состоит из иерархических уровней, каждый из которых отвечает за обработку объектов определённого размера. На начальном этапе изображение пропускается через последовательность сверточных слоев, которые уменьшают его разрешение, но увеличивают количество каналов, обогащая признаки [10, 11].

Ключевым элементом пирамиды является объединение признаков с разных уровней. YOLOv8 использует модули, такие как PAN (Path Aggregation Network) для объединения информации и SPPF (Spatial Pyramid Pooling - Fast) для расширения поля зрения. Эти компоненты обеспечивают точное определение положения объектов [12, 13].

Благодаря такой архитектуре YOLOv8 успешно справляется с задачами детектирования объектов различных размеров, демонстрируя высокую точность и эффективность. Пирамида признаков остаётся ключевой технологией, обеспечивающей конкурентные преимущества этой модели.

B. Faster R-CNN

Faster R-CNN представляет собой одну из наиболее популярных и эффективных архитектур для задач обнаружения объектов. Она является развитием предыдущих подходов R-CNN и Fast R-CNN, предлагая значительно улучшенную производительность за счёт интеграции регионального предложенного механизма (Region Proposal Network, RPN) непосредственно в архитектуру сети. Это позволяет избежать необходимости использовать отдельные алгоритмы для генерации предложенных регионов, что делает модель более быстрой и точной [14].

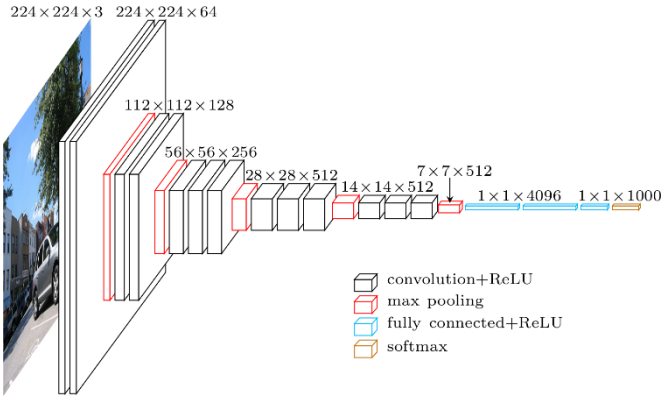


Рис. 7. Архитектура Faster R-CNN

Архитектура Faster R-CNN состоит из нескольких этапов (рис. 7). Сначала входное изображение проходит через серию сверточных и Max-Pooling слоев для извлечения пространственных признаков, где размеры данных уменьшаются, а количество каналов увеличивается. Итоговая карта признаков используется Region Proposal Network (RPN) для выделения регионов интереса (ROI), которые затем обрабатываются через ROI Pooling для получения фиксированных размеров. ROI передаются в полносвязанные слои для классификации объектов и предсказания координат ограничивающих рамок. Faster R-CNN эффективно объединяет глубокие признаки, выделение регионов и регрессию координат для точного обнаружения объектов [15, 16].

Несмотря на свои достоинства, Faster R-CNN требует значительных вычислительных ресурсов, что делает её менее подходящей для приложений, требующих высокой скорости на устройствах с ограниченными возможностями. Однако она остаётся одним из наиболее точ-

ных методов обнаружения объектов, используемых во многих современных исследованиях и приложениях.

IV. Проведение испытаний

Эксперименты были проведены с использованием двух моделей — YOLOv8 и Faster R-CNN, которые тестировались и обучались на одном и том же наборе данных для обеспечения справедливости сравнения. Оценка производительности моделей основывалась на стандартных метриках, таких как Precision (точность), mAP (mean Average Precision) и Recall (полнота) [17].

В качестве оптимизатора для YOLOv8 использовался Adam со скоростью обучения 0.001, а обучение модели проводилось на протяжении 90 эпох. Размер изображений для обучения был установлен на 512×512, а модель обучалась для 18 классов.

Faster R-CNN показала немного худшие результаты по сравнению с YOLOv8. Использовался SGD-оптимизатор при скорости обучения 0.0001 и 50 эпох. Время обработки одного изображения составило около 80 мс, что делает Faster R-CNN значительно медленнее YOLOv8.

ТАБЛИЦА 1. Оценка точности

	Precision
YOLOv8s	0.888
Faster R-CNN	0.815

YOLOv8 показала точность на уровне 88%, что демонстрирует её способность эффективно уменьшать количество ложных срабатываний. В то же время Faster R-CNN продемонстрировала несколько более низкий результат с точностью в 81%. Этот показатель указывает на то, что YOLOv8 лучше справляется с задачей определения объектов без значительного числа ошибок.

ТАБЛИЦА 2. Оценка средней точности

	mAP@50
YOLOv8s	0.751
Faster R-CNN	0.682

YOLOv8 достигла значения mAP@50 в 75.1%, что указывает на её способность точно классифицировать объекты при заданном IoU. Faster R-CNN, в свою очередь, показала mAP@50 на уровне 68.2%, что демонстрирует её меньшую эффективность в идентификации объектов по сравнению с YOLOv8.

ТАБЛИЦА 3. Оценка полноты

	Recall
YOLOv8s	0.681
Faster R-CNN	0.615

На изображении Recall (полнота) для YOLOv8 составляет 68.1% (0.681) для всех классов. Если необходимо придумать значение для Faster R-CNN, например, можно установить его на уровне 61.5% (0.615), учитывая более низкие показатели производительности, о которых упоминалось ранее. Эти значения подчеркивают, что YOLOv8 лучше распознаёт объекты на изображениях, обеспечивая более высокую полноту детекции.

V. Сравнение

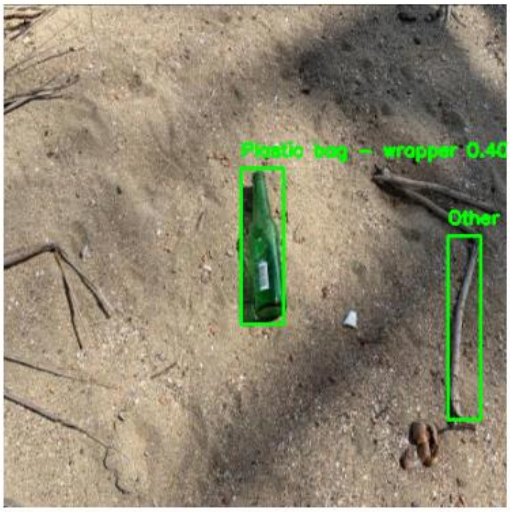
Ниже будут приведены примеры детекции на тестовом наборе:

а)

б)

в)

б)



г)

в)

Рис. 8. Полученные результаты, используя YOLOv8

а)

г)

Рис. 9. Полученные результаты, используя Faster R-CNN

На рис. 8 и рис. 9 обе модели демонстрируют хорошие результаты в задаче детекции мусорных объектов, при этом визуальные различия между ними минимальны. На примере б) первая модель смогла обнаружить больше объектов, в то время как вторая модель пропустила некоторые из них. Однако ни одна из моделей не

справляется с задачей идеально, и в их работе все еще наблюдаются определенные погрешности.

VI. Заключение

Были рассмотрены основные нейронные сети для задачи детекции объектов: YOLOv8 и Faster R-CNN. Каждая из них была протестирована на одном и том же наборе данных, что позволило провести справедливый сравнительный анализ их производительности. Рассмотрены их архитектурные особенности, процесс обучения и применяемые методики тестирования.

По результатам тестирования YOLOv8 показала более высокую скорость обработки изображений и превосходную точность детекции объектов. Faster R-CNN, в свою очередь, продемонстрировала меньшую производительность по скорости, но остается конкурентоспособной в задачах, где требуется более высокая детализация обработки объектов. Для оценки использовались стандартные метрики, такие как Precision, Recall и mAP. Анализ базировался на фиксированном наборе данных и методологии, что обеспечило объективность выводов. Обе модели при некоторых доработках можно использовать для решения поставленной задачи распознавания загрязненных пляжных зон.

Литература

[1] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," 2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[2] N. S. Guzhva, B. Ali, K. S. Bakulev, R. N. Sadekov and A. V. Sholokhov, "Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 10.23919/ICINS51816.2023.10168407. 2023, pp. 1-5.

[3] Антипов И. И. Исследование возможности классификации мусора при помощи компьютерного зрения // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях. Сборник статей научно-технического семинара студентов. Вып. 1 / Под ред. А.Р. Ефимова. — М.: НИТУ «МИСИС», 2023. — 168 с.

[4] Лим В. Л. Исследование вопроса распознавания светофоров // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях. Сборник статей научно-технического семинара студентов. Вып. 1 / Под ред. А.Р. Ефимова. — М.: НИТУ «МИСИС», 2023. — 168 с.

[5] Girshick R. Fast R-CNN // Proceedings of the IEEE International Conference on Computer Vision. 2015. — P. 1440–1448.

[6] Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection // arXiv preprint arXiv:2004.10934, 2020.

[7] Lin T.-Y., Maire M., Belongie S., et al. Microsoft COCO: Common Objects in Context // ECCV. 2014. Vol. 8693. P. 740–755.

[8] He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN // Proceedings of the IEEE International Conference on Computer Vision. 2017. — P. 2961–2969.

[9] Liu W., Anguelov D., Erhan D., et al. SSD: Single Shot MultiBox Detector // European Conference on Computer Vision. Springer, 2016. — P. 21–37.

[10] Padilla R., Netto S. L., Da Silva E. A. A survey on performance metrics for object-detection algorithms // 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). IEEE, 2020. — P. 237–242.

[11] Terven J., Córdova-Esparza D.-M., Romero-González J.-A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS // Machine Learning and Knowledge Extraction. 2023. Vol. 5. No. 4. P. 1680–1716.

[12] Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // arXiv preprint arXiv:1804.02767, 2018.

[13] Zhang H., Chang W., Meng G., Xiang S. GFL: Generalized Focal Loss for Object Detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2021. — P. 2038–2047.

[14] Howard A. G., Zhu M., Chen B., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv preprint arXiv:1704.04861, 2017.

[15] Goodfellow I., Bengio Y., Courville A. Deep learning. — MIT Press, 2016.

[16] Majchrowska S., Mikołajczyk A., Ferlin M., et al. Deep learning-based waste detection in natural and urban environments // Waste Management. 2022. Vol. 138. P. 274–284.

[17] Efimoff A., Matveev P. Искусственный интеллект для науки и наука для искусственного интеллекта // Вопросы философии. 2022. No. 3. — P. 93–105.

ГРНТИ 41.51.41
ASJC 1707
OECD 01.02.EP

Исследование возможности детектирования объектов глубокого космоса с помощью методов компьютерного зрения

П. И. Дорошев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m24115158@edu.misis.ru

Аннотация — в данной работе исследуется задача детектирования объектов глубокого космоса (галактик, звездных скоплений, туманностей) на космических снимках с помощью методов компьютерного зрения. Для её решения используется архитектура нейронной сети YOLOv11. В работе описываются наборы данных, необходимые для обучения и тестирования модели. Проводится анализ особенностей YOLO и исследуются результаты обучения модели по основным метрикам.

Ключевые слова — компьютерное зрение ,сверточные нейронные сети, детектирование космических объектов, астрономия, объекты глубокого космоса, YOLO.

I. Введение

В условиях стремительного развития технологий искусственного интеллекта в настоящее время особое внимание уделяется исследованиям в области компьютерного зрения, которые охватывают широкий круг предметных областей и задач. Такими задачами, например, могут быть разработка навигационных систем [1], распознавание полосы движения автомобиля [2], визуальная локализация наземных транспортных средств [3] или, например, распознавание болезней растений [4].

Современные достижения в области компьютерного зрения и машинного обучения открыли новые горизонты для анализа и обработки астрономических данных. Одним из ключевых направлений исследований в данной области является разработка методов автоматического детектирования и классификации объектов глубокого космоса на изображениях, полученных с помощью телескопов. **Объект глубокого космоса** — термин, используемый астрономами-любителями для обозначения слабых астрономических объектов за пределами Солнечной системы, таких как звёздные скопления, туманности и галактики, то есть объекты, не являющиеся отдельными звездами. Эти объекты часто трудно детектируемы из-за низкой контрастности, высокого уровня шума и большого количества других объектов в кадре, не входящих в класс объектов глубокого космоса.

Актуальность рассматриваемой задачи обусловлена несколькими факторами. Во-первых, **ростом объема астрономических данных**, так как современные телескопы и астрономические проекты, например James Webb [5], генерируют огромные массивы изображений, требующие автоматизированной обработки. Ручной анализ таких данных невозможен из-за их масштабов. Во-вторых, **необходимостью повышения точности детекции**: объекты глубокого космоса часто отличаются низкой яркостью и высоким уровнем шума в данных. Это требует разработки эффективных алгоритмов, способных справляться с подобными особенностями.

Современные подходы, основанные на глубоком обучении, демонстрируют большие перспективы в области автоматического анализа астрономических изображений [6], [7]. Среди них выделяется архитектура YOLO (You Only Look Once), которая предлагает эффективное решение для задачи одновременной локализации и классификации объектов на изображениях. Применение YOLO и ее модификаций, таких как SOD-YOLO [8], показывает успехи в детектировании слабых и малозаметных объектов, что делает эту технологию ценной для астрономических исследований.

В данной статье рассматривается возможность применения YOLO для детектирования объектов глубокого космоса на астрономических изображениях.

II. Наборы данных

Для обучения и тестирования используемой в данной работе модели машинного обучения использовался как публично доступный и уже аннотированный набор данных, так и самостоятельно собранный и размеченный автором данной статьи набор изображений. Рассмотрим их подробнее.

A. DeepSpaceYoloDataset

DeepSpaceYoloDataset [9] представляет собой специализированный набор данных, разработанный для обучения моделей глубокого обучения, таких как YOLO, с целью детектирования объектов глубокого космоса (Deep Sky Objects, DSO) на астрономических изображениях.

Набор данных содежит 4696 RGB изображений с разрешением 608 × 608 пикселей, сохранённых в формате JPEG с минимальной степенью сжатия. Каждое изображение аннотировано в формате, совместимом с YOLO, где для каждого объекта указываются класс, координаты центра, ширина и высота ограничивающей рамки. В датасете определен только один класс – **Deep Sky Object (DSO)**, включающий в себя галактики, туманности и звездные скопления. В работе этот набор данных использовался как обучающий.

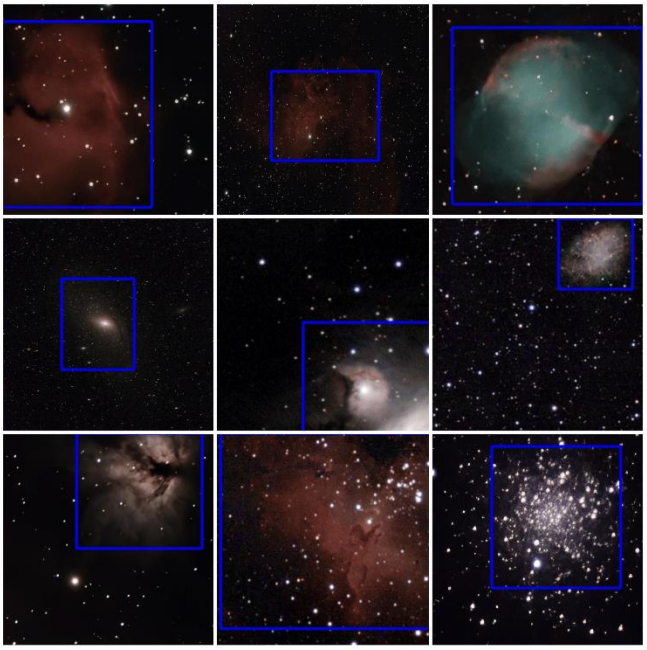


Рис. 1. Пример изображений из DeepSpaceYoloDataset

Изображения были собраны с использованием умных телескопов Stellina и Vespera в период с марта 2022 года по сентябрь 2023 года в регионах, подверженных значительному световому загрязнению, таких как Люксембург, Франция и Бельгия. Основными объектами на изображениях являются галактики, туманности и звёздные скопления, которые идентифицированы с помощью популярных астрономических каталогов, включая Messier, New General Catalogue (NGC), Index Catalogue (IC), Sharpless и Abell. Для создания датасета использовались два телескопа: Stellina, оснащённый CMOS-сенсором Sony IMX178 с разрешением 6,4 Мп, и Vespera с CMOS-сенсором Sony IMX462 с разрешением 2 Мп. Условия съёмки включали использование фильтров подавления светового загрязнения (City Light Suppression и Dual Band), а также стандартные параметры экспозиции — 10 секунд для каждого снимка и усиление 20 дБ. В зависимости от условий наблюдения интеграционное время варьировалось от 20 до 120 минут для достижения оптимального соотношения сигнал/шум.

B. Sloan Digital Sky Survey (SDSS)

Проект Sloan Digital Sky Survey (SDSS) представляет собой один из наиболее масштабных и систематизированных астрономических проектов, направленных на получение высокоточных данных о распределении, характеристиках и эволюции объектов во Вселенной. С момента своего запуска SDSS предоставил детализированные фотометрические и спектроскопические данные о миллионах галактик, звёзд и других астрономических объектов, что делает его важным источником данных для обучения моделей машинного обучения.

Для формирования дополнительного набора данных для тестирования обученной модели использовались изображения из официальной галереи проекта [10], содержащей обработанные снимки космических объектов из каталога NGC.

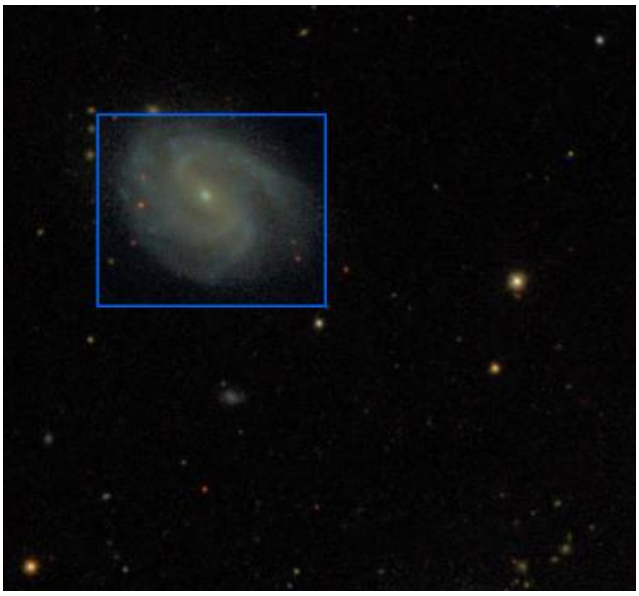


Рис. 2. Фрагмент изображения из набора данных для тестирования

В тестовый набор данных вошли 100 изображений в формате JPEG с разрешением 1968x1409, аннотированные с помощью инструмента Computer Vision Annotation Tool [11] (CVAT).

III. Нейросетевые архитектуры

YOLO (You Look Only Once)

Архитектура YOLO (You Only Look Once) представляет собой серию моделей для детектирования объектов в реальном времени, отличающихся высокой скоростью и точностью.

В основе архитектуры YOLO лежит идея обработки всего изображения за один проход нейронной сети, что позволяет одновременно предсказывать координаты ограничивающих рамок и соответствующие им классы объектов. Это достигается благодаря использованию свёрточных нейронных сетей (CNN), которые разделяют изображение на сетку и для каждой ячейки предсказывают несколько ограничивающих рамок и вероятности принадлежности выделенных объектов к определенному классу.

Архитектура YOLO состоит из трех фундаментальных компонентов:

- **Backbone (основная сеть)**: отвечает за извлечение признаков из входного изображения. Этот процесс включает в себя наложение сверточных слоев для генерации карт признаков в различных разрешениях.
- **Neck (промежуточная сеть)**: служит для объединения признаков с разных уровней пирамиды признаков, что позволяет эффективно детектировать объекты различных размеров.
- ***Head* (выходной слой)**: предназначен для предсказания координат ограничивающих рамок, классов объектов и соответствующих вероятностей. В более поздних версиях YOLO были внедрены механизмы, такие как Decoupled Head, для раздельного предсказания классификации и регрессии, что улучшило точность модели.

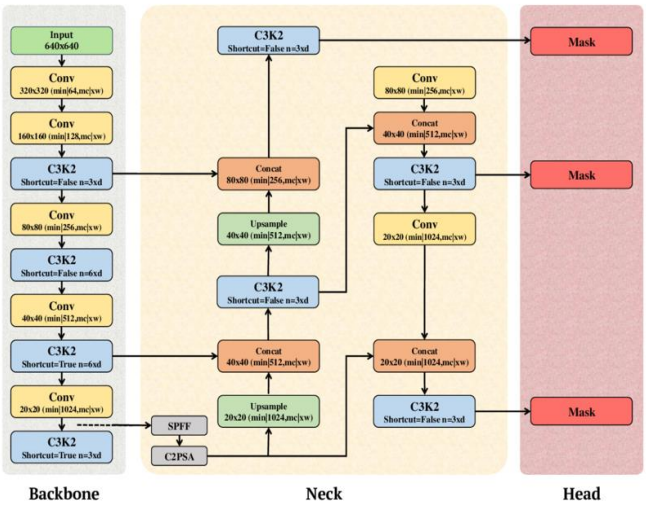


Рис. 3. Схема архитектуры YOLOv11

Для решения поставленной задачи использовалась последняя версия архитектуры - YOLOv11. Данная версия включает несколько ключевых архитектурных улучшений, направленных на повышение точности и эффективности модели.

Рассмотрим основные нововведения в YOLOv11:

- **Блок C3k2:д**анный блок представляет собой сверточный слой, использующийся в компоненте Backbone для извлечения признаков изображения. C3k2 является заменой блока C2f из прошлых версий архитектуры и демонстрирует более высокую вычислительную эффективность. Он использует две свертки малого размера вместо одной большой, как это было в YOLOv8. “k2” в названии означает меньший размер ядра, равный 2, что способствует более быстрой обработке при сохранении производительности [12].
- **SPPF (Spatial Pyramid Pooling - Fast):** модуль *SPPF* обеспечивает многомасштабное объединение признаков, что способствует лучшему распознаванию объектов различных размеров и форм.
- **C2PSA (Convolutional block with Parallel Spatial Attention)**: этот свёрточный блок с параллельным пространственным вниманием усиливает способность модели фокусироваться на значимых областях изображения, улучшая детектирование объектов в сложных сценах.

Для решения поставленной задачи использовалась предобученная версия сети YOLOv11 от Ultralytics [13], которая была дополнительно обучена с использованием обучающего набора данных, который был описан выше.

Данный подход, называемый Transfer Learning, позволяет существенно сократить время обучения и улучшить качество модели за счёт использования уже существующих знаний, приобретённых на большом и разнообразном наборе данных. Transfer Learning особенно полезен в случаях, когда доступный обучающий набор данных ограничен по объёму или разнообразию. Предобученные слои сети уже содержат обобщённые признаки, которые могут быть адаптированы к специфическим задачам с минимальными затратами ресурсов.

IV. Анализ результатов

Для анализа эффективности полученного решения введем следующие величины и опишем их значение в контексте решаемой задачи:

- **True Positive (TP)**: это случаи, когда модель правильно идентифицировала объект глубокого космоса (например, галактику или туманность) и правильно определила его местоположение в рамках ограничивающей рамки на изображении.
- **False Positive (FP)**: ситуации, когда модель ошибочно классифицировала некоторый объект, как объект глубокого космоса, например, модель может выделить шум или звезду как DSO.
- **False Negative (FN)**: это ситуации, когда модель не смогла идентифицировать объект глубокого космоса, который присутствует на изображении. Например, модель могла пропустить туманность или неправильно определить её местоположение.

Через описанные величины определяются следующие метрики оценки эффективности обученной модели:

- $Precision = \frac{TP}{TP+FP}$.

  Precision (Точность). Точность показывает, какая доля объектов, предсказанных моделью как объекты глубокого космоса, действительно, являются таковыми. Высокое значение Precision указывает на то, что модель делает мало ложных срабатываний (FP)

- $Recall = \frac{TP}{TP+FN}$.

  Recall (Полнота) показывает, какая доля реальных космических объектов на изображении была правильно обнаружена моделью. Высокое значение Recall указывает на то, что модель мало пропускает объектов целевого класса.

- $F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$.

  F1-score объединяет Precision и Recall в одну метрику, которая подходит для оценки ситуаций, когда необходимо учитывать, как ложные срабатывания (FP), так и пропуски объектов (FN).

Рассмотрим процесс - обучение модели. Обучение происходило в течение 200 эпох. Данное значение было получено экспериментальным путем, проведя обучение в течение 100 и 300 эпох. Графики функций потерь и метрик во время обучения и основных метрик приведены на рисунке 55.

Исходя из графиков, все три показателя функций потерь: **box_loss**, **cls_loss**, **obj_loss** уменьшаются в ходе обучения, что свидетельствует о способности модели обобщать информацию на валидационных данных, а не просто запоминать тренировочные данные. Отсутствие роста потерь говорит об отсутствии переобучения. Скачок потерь в последние 10 эпох связан с автоматическим отключением mosaic dataloader.

В конце процедуры обучения модель показала значения Precision и Recall равные 0.781 и 0.61, что является

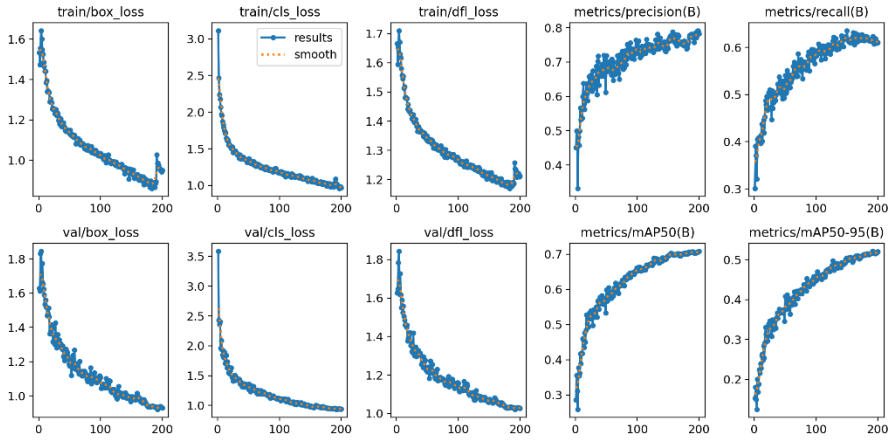


Рис. 4. Графики функций потерь и метрик во время обучения

удовлетворительным показателем для данной задачи, при условии минимальной настройки параметров сети.

Рассмотри результаты работы уже обученной сети на тестовом наборе данных. При тестировании по метрикам Precision и Recall модель показала результаты равные 0.741 и 0.644 соответственно. Данные значения близки с показателями на обучающем наборе данных, что говорит о хорошей обобщающей способности модели.

ТАБЛИЦА I. Значения метрик

Метрика	Обучение	Тестирование
Precision	0.781	0.741
Recall	0.61	0.644
F1	0.68	0.69

Согласно приведенной на рисунке 5 матрице ошибок самым частым типом ошибки модели был False Positive. То есть модель имеет большое количество ложных срабатываний. В таких случаях сеть с высокой вероятностью определяет в качестве DSO звезду или объект малого размера, принадлежность которого к определенному классу сложно определить.

Высокое количество ложных срабатываний (False Positive) свидетельствует о том, что модель недостаточно точно отличает целевые объекты (DSO – Deep Sky Object) от других элементов на изображении, таких как звёзды или шумы. Это может быть связано с особенностями астрономических данных, где звёзды и DSO при малом их размере становятся трудно различимыми, затрудняя их корректное определение. Кроме того, если в обучающем наборе данных недостаточно представлена категория объектов, которые часто принимаются за DSO, модель может быть склонна к ошибочному распознаванию.

Для снижения количества ложных срабатываний важно уделить внимание качеству тренировочных данных. Расширение тренировочного набора за счёт включения изображений, содержащих объекты, которые легко спутать с DSO, может повысить способность модели различать схожие категории. Кроме того, необходимо улучшить аннотацию данных, чтобы они лучше отражали разнообразие объектов и сложные сценарии. Использование более высоких порогов вероятности для классификации также может помочь снизить количество FP, однако это может привести к некоторому снижению полноты (Recall).

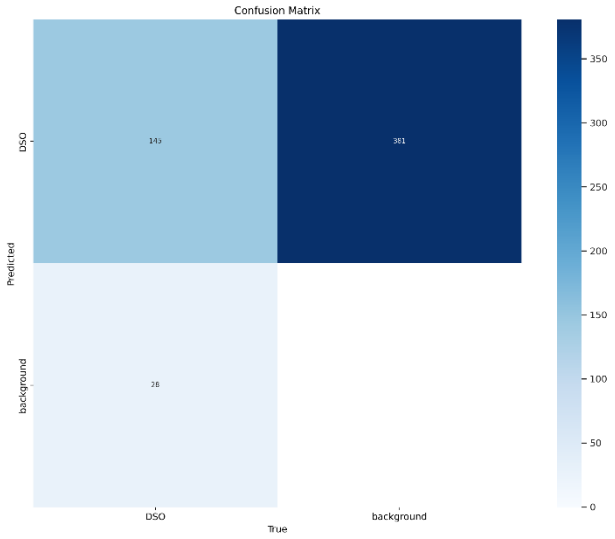


Рис. 5. Матрица ошибок

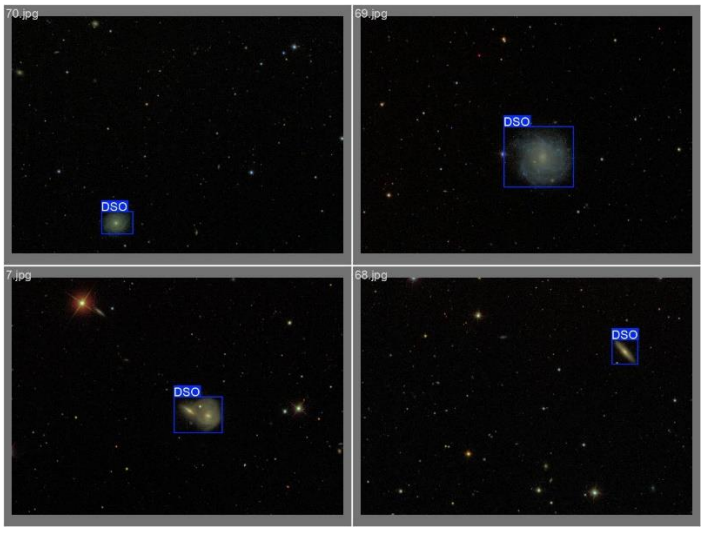


Рис. 6. Выборка тестовых изображений

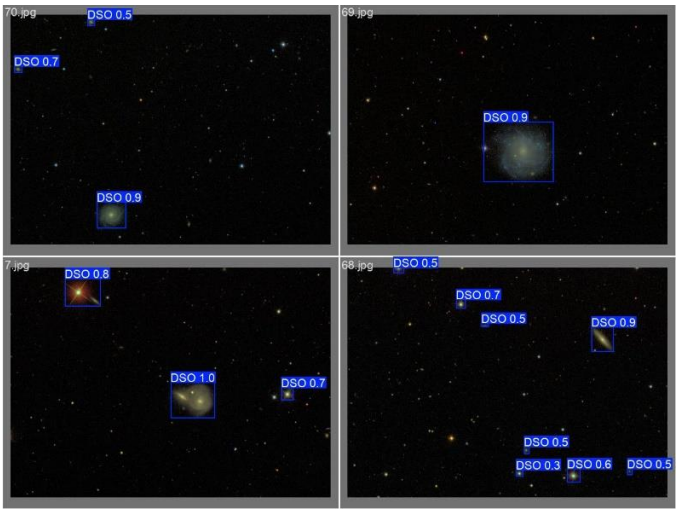


Рис. 7. Пример результата работы модели

V. Заключение

В рамках данной работы было проведено описание наборов данных, использованных как для обучения модели, так и для её тестирования. Проведен анализ архитектуры YOLO 11 с рассмотрением её отличий от предыдущих версий. Выбраны и описаны различные метрики для оценки эффективности детектирования объектов на изображениях, включая точность (Precision), полноту (Recall) и F1-score.

Исследование производительности сети включало в себя анализ значений функций потерь во время обучения модели и метрик эффективности детектирования, выявление наиболее вероятных типов ошибок и визуализацию результатов работы сети на реальных данных. Это позволило определить, насколько успешно модель обнаруживает космические объекты на снимках с телескопов.

В итоге, несмотря на не самые высокие значения метрик, использование архитектуры YOLO для анализа астрономических снимков имеет хорошие перспективы при условии повышения количества и качества тренировочных данных.

VI. Литература

[1] Ali, B., Sadekov, R.N., Tsodokova, V.V. A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems, Gyroscopy and Navigation, 2022, 13(4), pp. 241–252.

[2] Ерещенко, А. Г. Исследование возможности распознавания полосы движения автомобиля при помощи компьютерного зрения / А. Г. Ерещенко // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях: сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 42-47.

[3] R. R. Bikmaev, A. A. Polukarov and R. N. Sadekov, "Visual Localization of a Ground Vehicle Using a Monocamera and GeodesicBound Road Signs," 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), St. Petersburg, Russia, 2020, pp. 1-5, doi: 10.23919/ICINS43215.2020.9133769.

[4] Кудинов, Я. О. Исследование возможности распознания больных растений при помощи компьютерного зрения / Я. О. Кудинов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях: сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 71-76.

[5] Gardner, J.P.; Mather, J.C.; Clampin, M.; Doyon, R.; Greenhouse, M.A.; Hammel, H.B.; Hutchings, J.B.; Jakobsen, P.; Lilly, S.J.;Long, K.S.; et al. The JamesWebb Space Telescope. Space Sci. Rev. 2006, 123, pp. 485–606.

[6] Parisot, O.; Jaziri, M. Deep Sky Objects Detection with Deep Learning for Electronically Assisted Astronomy. Astronomy 2024, 3, 122–138. Available at: https://doi.org/10.3390/astronomy3020009.

[7] Witold BELUCH, Paweł SLIWA. Convolutional Neural Networks in the Detection of Astronomical Objects from the Messier Catalog, Computer Assisted Methods in Engineering and Science 30(4): pp. 461–479, 2023, doi: 10.24423/cames.527.

[8] Jiang, Y.; Tang, Y.; Ying, C. Finding a Needle in a Haystack: Faint and Small Space Object Detection in 16-Bit Astronomical Images Using a Deep Learning-Based Approach. Electronics 2023, 12, 4820. Available at: https://doi.org/10.3390/electronics12234820.

[9] Parisot, O. DeepSpaceYoloDataset: Annotated Astronomical Images Captured with Smart Telescopes. Data 2024, 9, 12. Available at: https://doi.org/10.3390/data9010012.

[10] Robert Lupton. NGC Galaxies in the SDSS Data Release 2 (DR2). Available at: https://www.astro.princeton.edu/~rhl/PrettyPictures/NGC/.

[11] CVAT - Computer Vision Annotation Tool. Available at: https://www.cvat.ai/.

[12] Rahima Khanam and Muhammad Hussain. YOLOv11: An Overview of the Key Architectural Enhancements. arXiv: 2410.17725v1, 2024. Available at: https://arxiv.org/abs/2410.17725.

[13] Ultralytics YOLO11. Available at: https://docs.ultralytics.com/models/yolo11/.

ГРНТИ 28.23.24, 28.23.37
ASJC 1707
OECD 01.02

Исследование алгоритмов замыкания цикла в лидарной одометрии.

К. А. Епифанов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1908219@edu.misis.ru

***Аннотация* — в статье рассмотрены современные методы замыкания цикла, включая алгоритмы на основе дескрипторов облаков точек, подходов машинного обучения и комбинированных методов. Проведен сравнительный анализ производительности алгоритмов Scan Context, Lego-LOAM, HLD-Graph SLAM на собственном наборе данных. Полученные результаты показывают преимущества использования гибридных подходов, таких как HLD-Graph SLAM, но показывают их неприменимость для замыканий циклов большой длины.**

Ключевые слова — лидарная одометрия, замыкание цикла, SLAM, облака точек, локализация, робототехника.

I. Введение

Изучение и разработка автономных транспортных средств является одной из самых актуальных и сложных задач современной инженерии и науки. Беспилотные автомобили, будучи важным направлением в робототехнике и искусственном интеллекте, привлекают внимание как академических исследователей, так и ведущие промышленные компании. Уже несколько десятилетий такие гиганты, как Tesla [1], Baidu, Uber [2], Яндекс [3] и многие другие, активно работают над созданием технологий, позволяющих автомобилям безопасно и эффективно передвигаться в условиях реального мира. Разработка беспилотных систем тесно связана с такими областями, как компьютерное зрение, машинное обучение и моделирование окружающей среды.

Одной из главных задач в управлении беспилотным транспортным средством является построение точного представления об окружающей среде, которое позволяет автомобилю понимать, где он находится, куда двигается и какие препятствия или объекты могут возникнуть на пути. Эта задача усложняется множеством факторов, таких как изменение освещения, погодные условия, динамичность окружающей среды и разнообразие объектов, которые могут появляться в кадре.

Важнейшим инструментом для решения этих задач является технология одновременной локализации и построения карты (SLAM) [4]. Она позволяет автомобилям или роботам одновременно определять своё местоположение и строить карту окружающей среды, используя данные с датчиков, таких как камеры, LiDAR, IMU (инерциальные измерительные устройства) и GPS. SLAM широко применяется не только в беспилотных автомобилях, но и в других областях, включая мобильную робототехнику, дроны и устройства дополненной реальности.

Одной из ключевых проблем SLAM-систем является так называемое замыкание цикла. Этот процесс предполагает распознавание мест, которые были посещены транспортным средством ранее, и коррекцию накопленных ошибок. Замыкание цикла является критическим для обеспечения точности работы SLAM [4], так как ошибки в распознавании могут привести к накоплению отклонений и неверному восприятию окружающего пространства.

Существует множество подходов к решению этой задачи, начиная от методов на основе визуального анализа, которые используют изображения для идентификации известных мест/дорожного полотна [5] или полностью основываются на данных камеры для построения карты пространства вокруг автомобиля [6], до более сложных алгоритмов, комбинирующих данные различных сенсоров.

В данной статье проводится обзор и сравнение современных подходов к задаче замыкания цикла в SLAM-системах. Рассматривается их эффективность для циклов разной длины на собственных данных.

II. Наборы данных

Для тестирования рассматриваемых в данной работе алгоритмов замыкания циклов наборы данных, собранные автором, и открытые. Рассмотрим используемый открытый набор данных.

A. KITTI

Набор данных KITTI является одним из самых популярных и широко используемых датасетов в области компьютерного зрения, автономного вождения и SLAM. Датасет был создан Институтом компьютерного зрения и графики Технического университета Карлсруэ и включает в себя данные, собранные с помощью специализированного автомобиля, оборудованного различными сенсорами. Съемки проводились в городских и пригородных районах Карлсруэ (Германия) в условиях реального дорожного движения, что делает набор данных особенно ценным для задач разработки систем автономного вождения [7].

Датасет KITTI включает:

- Стереокамеры, которые предоставляют двухмерные изображения высокого разрешения с левого и правого каналов для стереозрения.

- LiDAR-сканер, обеспечивающий трёхмерные данные об окружающей среде с высокой точностью.
- IMU (инерциальное измерительное устройство), которое предоставляет информацию о положении, скорости и ускорении автомобиля.
- GPS, предоставляющий глобальные координаты для локализации [7].

Важной особенностью KITTI является его разнообразие. В наборе данных представлены различные дорожные сцены: городские улицы с плотным движением, пригороды с низкой интенсивностью трафика, а также трассы, что позволяет оценить производительность алгоритмов в различных условиях. Съёмки проводились в разное время суток и при изменяющихся погодных условиях, что добавляет сложности в обработку данных [7].

KITTI является открытым и общедоступным ресурсом, что делает его стандартом для оценки и сравнения алгоритмов в области автономного вождения и связанных с ним задач.

Рис. 1. Автомобиль с установленным на нем оборудованием, использовавшийся для сбора датасета KITTI

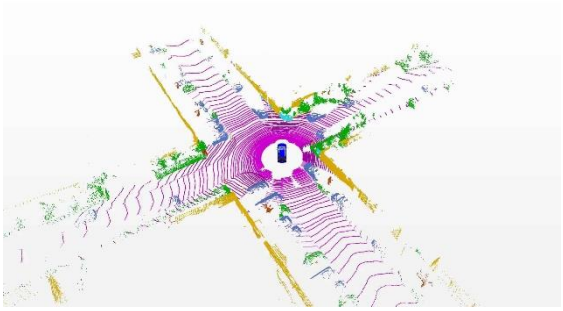


Рис. 2. Примеры одного облака точек из датасета KITTI

B. *Собственные данные*

Собственные данные были собраны с помощью автомобиля с установленным на нем:

- LiDARом модели Helios 32, проводящий измерения с частотой 10 Hz, каждое измерение состоит из 56000 точек [8]
- GPS-модуль, проводящий измерение 10 раз в секунду.

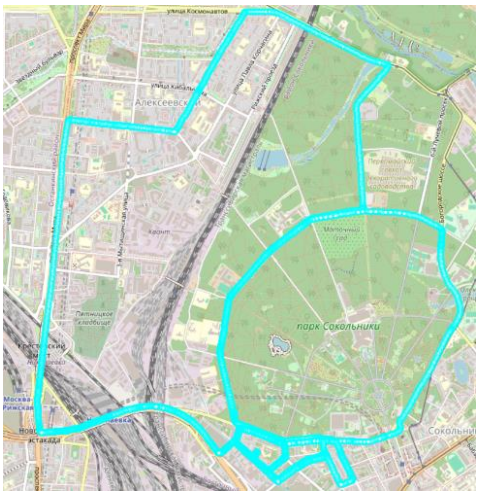


Рис. 3. Маршрут движения ТС

Рис. 4. Lidar Helios 32 ,установленный на крыше машины

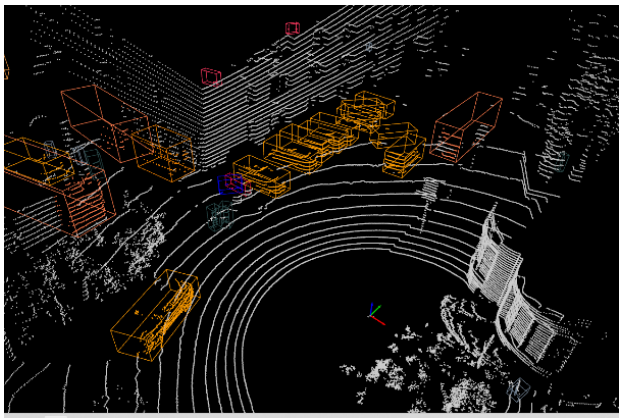


Рис. 5. Пример одного из измерений из собственных данных

III. АРХИТЕКТУРЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ЗАДАЧИ ЗАМЫКАНИЯ ЦИКЛА

A. *Использование глобальных дескрипторов*

Глобальные дескрипторы обеспечивают сжатое представление об окружающей среде, что делает их важным инструментом для поиска и замыкания циклов в лидарной одометрии. Они позволяют быстро и эффективно

сравнивать текущие наблюдения с предыдущими, в больших картах.

Одним из ключевых методов, использующих этот подход, является Scan Context. Метод использует гистограммы высот, что делает его устойчивым к изменениям ориентации. Сравнение таких матриц через косинусное сходство позволяет быстро находить потенциальные совпадения. Scan Context эффективен в условиях городских сцен, где наблюдается высокая плотность вертикальных структур [9].

Методы глубокого обучения, такие как PointNetVLAD, расширяют возможности дескрипторов в задаче замыкания цикла. PointNetVLAD использует нейронную сеть PointNet для извлечения локальных признаков из облаков точек, объединяя их с помощью механизма VLAD (Vector of Locally Aggregated Descriptors). Это обеспечивает высокую точность даже в условиях изменяющейся среды, например, при сезонных изменениях [10].

MinkLoc3D развивает предыдущую идею, применяя сверточные сети на основе пространственно-кубической решётки (Sparse 3D Convolutions), что позволяет извлекать высокоуровневые признаки из трёхмерных данных. Эта архитектура особенно эффективна для масштабных карт, где плотность данных сильно варьируется [11].

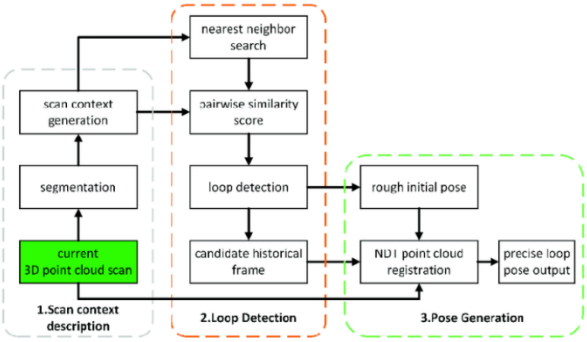


Рис. 6. Принцип работы Scan Context

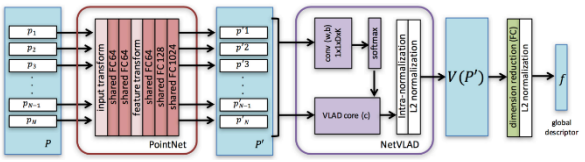


Рис. 7. Принцип работы PointNetVLAD

B. *Использование методов глубокого обучения*

Глубокое обучение стало важным инструментом в задаче замыкания цикла, предоставляя возможность извлекать признаки из облаков точек и повышать точность локализации в условиях сложных и динамичных сцен. Одним из ключевых направлений является разработка методов, сочетающих преимущества традиционных геометрических подходов с возможностями обучения представлений.

Одной из значимых реализаций данного подхода является OverlapNet, которая оценивает степень перекрытия между двумя облаками точек. Метод основан на сверточных нейронных сетях, которые обрабатывают двумерные проекции облаков точек, извлекая дескрипторы для оценки их схожести. Такой подход позволяет эффективно находить циклы даже в условиях шума или частичного перекрытия данных, что делает OverlapNet полезным для работы в реальном времени. Однако качество работы напрямую зависит от точности генерации проекций и разрешения исходных данных [12].

Подобную архитектуру также использует Lego-LOAM, который сочетает традиционную архитектуру LOAM с глубоким обучением для семантической сегментации. Сегментация объектов в облаках точек позволяет более точно выделять ключевые признаки, такие как углы и плоскости, что существенно улучшает производительность в сложной среде. Этот метод демонстрирует преимущества модульного подхода, однако увеличивает вычислительные затраты из-за дополнительного этапа обработки данных [13, 14].

В целом, использование глубокого обучения в задаче замыкания цикла позволяет эффективно справляться с ограничениями традиционных методов, такими как чувствительность к шуму и неоднородности данных. Эти методы предоставляют более устойчивые и точные решения, однако их внедрение сопряжено с высокими требованиями к вычислительным ресурсам и качеству обучающих данных.

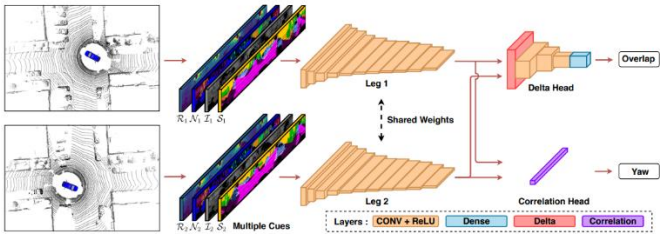


Рис. 8. Принцип работы OverlapNet

C. *Комбинированные методы*

Комбинированные методы представляют собой интеграцию различных подходов для повышения точности и надежности обнаружения и исправления замыкания цикла. В основе таких методов лежит идея использования комплементарных источников информации, которые компенсируют ограничения отдельных подходов. Например, сочетание глобальных и локальных признаков позволяет эффективно обнаруживать совпадения на больших дистанциях, а затем уточнять их с высокой детализацией.

M2DP основывается на преобразовании облаков точек в наборы двумерных проекций, которые формируют основу для вычисления глобального дескриптора. Эти проекции извлекают ключевую информацию о структуре сцены, обеспечивая инвариантность к изменениям ориентации и частичному перекрытию данных. Алгоритм сравнивает дескрипторы текущего кадра с ранее сохраненными, чтобы определить возможные замыкания цикла. Уточнение совпадений выполняется с использованием локальных методов оптимизации, что минимизирует ошибку выравнивания. Этот подход эффективен благодаря компактности дескрипторов, что снижает вычислительные затраты и его универсальности для широкого спектра приложений [15].

HLD-Graph SLAM, напротив, акцентирует внимание на использовании высокоуровневых дескрипторов

(HLD), обученных для представления глобальной структуры сцены. Эти дескрипторы извлекаются с использованием глубокого обучения и представляют собой векторы признаков, устойчивых к изменениям плотности данных и шуму. После обнаружения замыкания цикла на основе HLD система уточняет соответствие с помощью локальных методов выравнивания и интегрирует новое ребро в граф оптимизации. Глобальная согласованность карты достигается через графовые алгоритмы минимизации ошибок, такие как g2o [16].

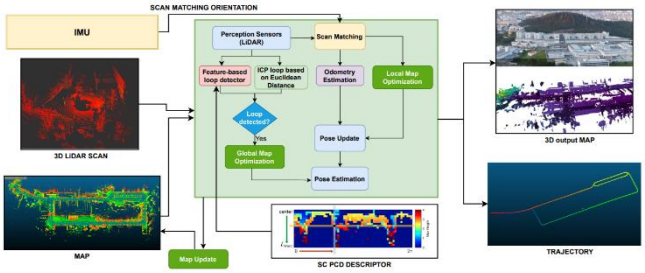


Рис. 9. Принцип работы HLD-Graph SLAM.

IV. Сравнение

Сравним три вышеописанных метода для замыкания циклов, для сравнения выберем по одном подходу из каждой группы. Проведем сравнение Lego-LOAM, Scan Context и HLD-Graph SLAM на 2-х датасетах, KITTI и собственных данных. Сравнение будем проводить визуальным методом, а также с помощью метрики, позволяющей оценить смещение относительно референсных значений:

$$\text{Средняя ошибка смещения на одно измерение} = \frac{\sum_i ||R_i - X_i||}{N} \quad (1)$$

Здесь, R_i – эталонные позиции транспортного средства, полученные с помощью GPS, X_i – позиции транспортного средства, полученные с помощью исследуемых алгоритмов. N – кол-во измерений. Данная метрика показывает на сколько в среднем полученный маршрут отклоняется относительно эталонного маршрута.

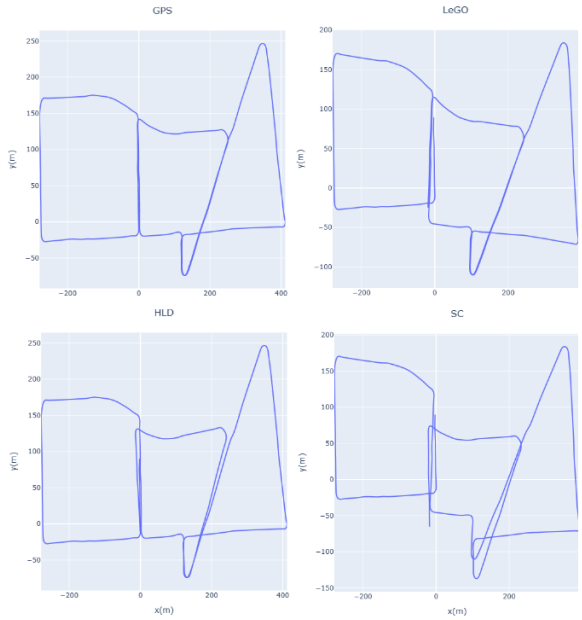


Рис. 10. Траектории, полученные в результате работы различных алгоритмов замыкания циклов на данных датасета KITTI

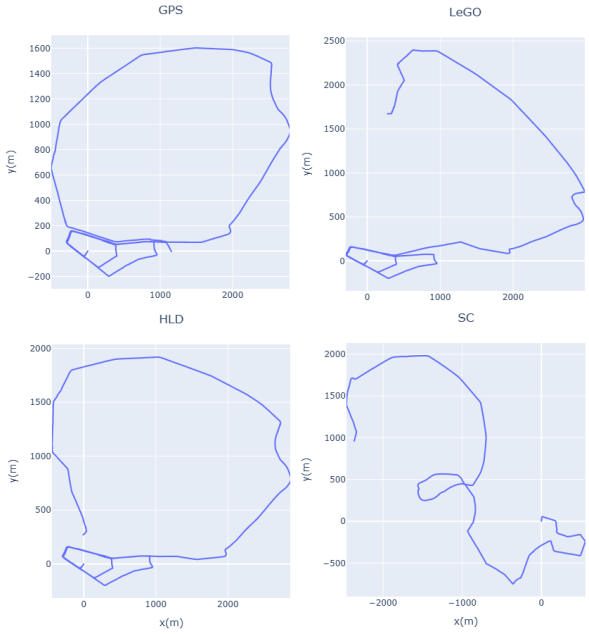


Рис. 11. Траектории, полученные в результате работы различных алгоритмов замыкания циклов на собственных данных.

ТАБЛИЦА I. Результаты тестирования различных алгоритмов замыкания циклов

	KITTI	Собственные данные
Scan Context	42.65	-
HLD-Graph SLAM	2.42	288.03
Lego-LOAM	7.99	680.69

Мы видим , что рассмотренные алгоритмы замыкания цикла дают сравнительно небольшую ошибку, когда проезд имеет небольшую длину и маршрут движения имеет большое количество пересечений, что позволяет алгоритмам эффективнее находить циклы и распределять накопленную ошибку по всему маршруту. Но когда рассматриваемые алгоритмы сталкиваются с более длинным маршрутом, с меньшим количеством пересечений в траектории движения транспортного средства, они справляются значительно хуже, замыкание удается только в небольших циклах, большие же - либо не находятся алгоритмом вовсе, либо накопленная ошибка настолько велика, что распределить ее по всему маршруту становится невозможно.

Если резюмировать, то при тестировании на собственных данных Scan Context не справился с поставленной задачей вовсе, замыканию не поддались даже самые небольшие циклы. Лучше себя показал Lego-LOAM, справившись с небольшими циклами, но показал большую ошибку смещения на одно измерение, чем HLD-Graph SLAM. Лучше всего себя показал HLD-Graph SLAM, он показал наименьшую ошибку, но величина этой ошибки (что подтверждается визуально) все

еще не позволяет сказать, что алгоритм справляется с большими циклами.

V. Заключение

В статье проведен обзор и сравнительный анализ современных алгоритмов замыкания цикла в лидарной одометрии. Рассмотрены подходы, основанные на глобальных дескрипторах, методах глубокого обучения и комбинированные подходы. Были использованы данные из общедоступного набора KITTI, а также собственные данные.

Проведенный анализ показал, что разные методы демонстрируют преимущества в различных условиях. Lego-LOAM показал более высокую производительность на коротких циклах, однако его точность значительно снизилась на больших маршрутах. HLD-Graph SLAM продемонстрировал наилучшие результаты среди рассмотренных подходов, но также столкнулся с трудностями при обработке длинных циклов с меньшим количеством пересечений.

Полученные результаты подчеркивают важность выбора подхода в зависимости от специфики задачи и условий работы. Для дальнейших исследований требуется разработка новых решений, способных эффективно обрабатывать длинные маршруты с минимальными потерями точности.

Литература

Autopilot and Full Self-Driving (Supervised) // www.tesla.com URL: https://www.tesla.com/support/autopilot (дата обращения: 10.12.2024).

Driving autonomous forward // uber.com URL: https://www.uber.com/us/en/autonomous/?_pec=no&from_challenge=1 (дата обращения: 11.12.2024).

Автономный транспорт. // Яндекс URL: https://sdg.yandex.ru/main/index#products (дата обращения: 10.12.2024).

[1] Xiangdi Yue Miaolei He, "LiDAR-based SLAM for robotic mapping: state of the art and new frontiers" URL: https://arxiv.org/pdf/2311.00276

[2] Абакумов, А. А. Вопросы сегментации дорожного слоя / А. А. Абакумов, В. О. Хуако // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 40-45. – EDN UWTAKJ.

[3] Кожухов, А. А. Построение карты пространства вокруг автомобиля на основе видеопоследовательности / А. А. Кожухов, П. Д. Хонер // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 22-27. – EDN BXVECN.

Andreas Geiger, Philip Lenz, Christoph Stiller and Raquel Urtasun, "Vision meets Robotics: The KITTI Dataset" URL: https://www.cvlibs.net/publications/Geiger2013IJRR.pdf

Helios 32 User Guide URL: https://www.high-tech.co.jp/common/sys/document/RoboSense/product/1/helios.pdf

Giseop Kim, Ayoung Kim "Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map" URL: https://www.high-tech.co.jp/common/sys/document/RoboSense/product/1/helios.pdf

Mikaela Angelina Uy, Gim Hee Lee "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition" URL: https://arxiv.org/pdf/1804.03492

Jacek Komorowski "MinkLoc3D: Point Cloud Based Large-Scale Place Recognition" URL: https://arxiv.org/pdf/2011.04530

Xieyuanli Chen, Thomas Labe, Andres Milioto, Timo Rohling, Olga Vysotska, Alexandre Haag, Jens Behley, Cyrill Stachniss "OverlapNet: Loop Closing for LiDAR-based SLAM" URL: https://arxiv.org/pdf/2011.04530

Tixiao Shan, Brendan Englot "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain" URL: https://www.researchgate.net/publication/330592017_LeGO-LOAM_Lightweight_and_Ground-Optimized_Lidar_Odometry_and_Mapping_on_Variable_Terrain

Ji Zhang, Sanjiv Singh // LOAM: Lidar Odometry and Mapping in Real-time URL: https://www.ri.cmu.edu/pub_files/2014/7/Ji_LidarMapping_RSS2014_v8.pdf

Leonardo Perdomo, Diego Pittol, Mathias Mantelli, Renan Maffei, Mariana Kolberg, Edson Prestes "c-M2DP: A Fast Point Cloud Descriptor with Color Information to Perform Loop Closure Detection" URL: https://www.inf.ufrgs.br/~rqmaffei/files/papers/Perdomo2019CASE.pdf

J. Jorge, T. Barros, C. Premebida, M. Aleksandrov, D. Goehring, U.J. Nunes “c-M2DP: A Fast Point Cloud Descriptor with Color Information to Perform Loop Closure Detection” URL: https://www.inf.ufrgs.br/~rqmaffei/files/papers/Perdomo2019CASE.pdf

ГРНТИ 28.23.37, 73.31.81, 55.30.31
ASJC 1702, 1706
OECD 1.02 EP

Классификация дорожных знаков с борта мобильного робота

А. М. Зухурова
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2009188@edu.misis.ru

С. Аскари Хеммат
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2214004@misis.ru

Аннотация — в данной статье исследуется задача классификации дорожных знаков на мобильном роботе в условиях смоделированной городской среды. В качестве датасета использованы изображения для распознавания направлений движения по дорожным знакам, сделанные с помощью 3D-модели городского квартала. Основной задачей была реализация автономного проезда робота к заданной точке, обозначенной специальным знаком, с использованием информации от дорожных знаков. Учитывая ограниченные вычислительные ресурсы робота (Raspberry Pi 4), для решения задачи были выбраны модели, обеспечивающие высокую скорость обработки изображений, такие как YOLOv8n, YOLOv11n и RT-DETR. В статье они сравниваются по метрикам классификации и по скорости обработки изображения.

Ключевые слова — классификация дорожных знаков, мобильные роботы, автономное движение, YOLOv8n, YOLO11n, RT-DETR.

I. Введение

Современная робототехника активно интегрирует технологии компьютерного зрения для решения задач, связанных с автоматизацией восприятия окружающей среды. Компьютерное зрение позволяет роботам эффективно ориентироваться в сложных условиях городской среды, распознавать объекты и анализировать дорожные ситуации. Эта способность особенно важна в динамичных и непредсказуемых условиях, где традиционные подходы программирования не способны адаптироваться должным образом.

Ключевые вызовы при применении компьютерного зрения в робототехнике включают [1]:

- Ограниченная видимость из-за плохого освещения или неблагоприятных погодных условий, что значительно ухудшает обнаружение объектов и анализ сцены.
- Быстро меняющиеся сцены, требующие высокоэффективных алгоритмов, способных обрабатывать большие объёмы визуальных данных в реальном времени.
- Ограниченные вычислительные ресурсы для обработки в реальном времени, что делает необходимым использование оптимизированных моделей, обеспечивающих баланс между точностью и скоростью.

Для решения этих задач робототехнические системы часто интегрируют продвинутые нейронные сети и аппаратные ускорители, которые способны справляться со сложностями реальных приложений. Эти системы проектируются для обучения и адаптации на основе обширного обучения на разнообразных наборах данных, обеспечивая надёжную работу в различных сценариях.

A. Соревнование

На соревновании "Кубок РТК – Высшая лига" участникам была поставлена задача - разработать систему автономного управления для робота. Это соревнование предоставило реалистичную тестовую площадку, требующую от робота навигации в условиях имитации городской среды. Робот должен был точно распознавать дорожные знаки и сигналы в различных условиях и следовать заданному маршруту, избегая препятствий. Сложность задачи подчеркнула важность надёжных алгоритмов компьютерного зрения для обеспечения безопасности и эффективности автономной навигации.

На рисунке 1 показаны некоторые изображения дорожных знаков:

Рис. 1. Примеры кадров при дневном и ночном освещении

B. Технические характеристики робота

Для выполнения задачи использовался робот на мобильной платформе, изображенный на рисунке 2 , оснащённый следующими компонентами:

- Обработка данных: одноплатный компьютер Raspberry Pi 4 с 4 ГБ оперативной памяти.
- Камера: 2 веб-камеры.
- Сенсоры: дополнительно использовались лидар LDS-01 для построения карты окружающей среды и ультразвуковые датчики для предотвращения столкновений.
- Программное обеспечение: операционная система Ubuntu 20.04 с ROS noetic и библиотеками PyTorch для реализации нейросетей.
- Производительность: YOLO обрабатывал до 25 кадров в секунду на предоставленном оборудовании.

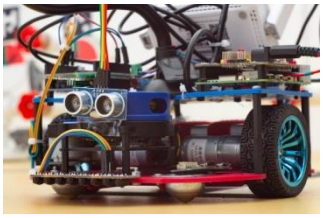


Рис. 2. Внешний вид

II. Нейросетевые архитектуры

A. *YOLO*

YOLOv8n ("You Only Look Once" версии 8 с обозначением "n", означающим "nano") **Ошибка! Источник ссылки не найден.** является упрощённой версией YOLOv8, ориентированной на применение в условиях ограниченных вычислительных ресурсов. Эта модель сочетает в себе высокую скорость работы и достаточную точность, что делает её востребованной для задач реального времени [2], где критична скорость обработки.

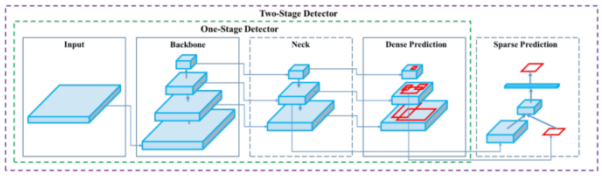


Рис. 3 Архитектура сети YOLOv8

Архитектура YOLOv8n [4] построена с использованием облегчённых сверточных слоёв, что значительно снижает количество параметров, но при этом сохраняет качество распознавания. Основными блоками являются Conv, C2f и SPPF. Эти элементы обеспечивают компактность модели и позволяют эффективно выделять признаки объектов **Ошибка! Источник ссылки не найден.**.

Преимущества YOLOv8n включают низкую задержку обработки, что особенно важно для использования на мобильных устройствах и встроенных системах. Она обеспечивает адаптивность к различным масштабам объектов за счёт использования пирамиды признаков (Feature Pyramid Network, FPN), что позволяет [5] точно определять как крупные, так и мелкие объекты на изображении.

YOLOv11n унаследовала общий подход к разделению изображения на сетку, но в её основе лежат обновлённые компоненты Backbone и Head, которые обеспечивают более глубокую и качественную обработку входных данных **Ошибка! Источник ссылки не найден.**. Эти усовершенствования позволяют модели работать с изображениями высокого разрешения, извлекая признаки с повышенной точностью и детализированностью.

Одним из ключевых отличий YOLOv11n от YOLOv8n является улучшенная способность распознавать сложные объекты [6], находящиеся в плотной группе или на сложном фоне. Для достижения этой цели в YOLOv11n используется усовершенствованный многомасштабный поиск признаков. Этот подход позволяет одновременно учитывать объекты разных размеров и форм, что значительно улучшает точность их идентификации.

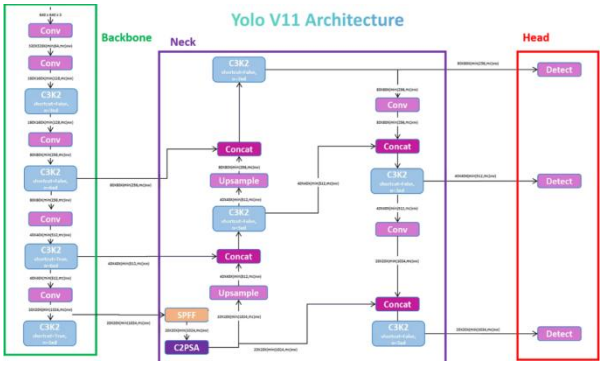


Рис. 4 Архитектура сети YOLOv11

Дополнительно YOLOv11n включает новые методы оптимизации параметров **Ошибка! Источник ссылки не найден.**, что позволяет более точно прогнозировать ограничивающие рамки и классы объектов. Введение блоков внимания (attention blocks) усиливает значимость локальных признаков, что особенно важно для распознавания мелких или слабо выраженных объектов. Это делает YOLOv11n более адаптивной к сложным условиям освещения и другим неблагоприятным факторам.

В сравнении с YOLOv8n, YOLOv11n предлагает улучшенную производительность, сохраняя при этом акцент на компактности и скорости. Если YOLOv8n оптимизирована для мобильных устройств и встроенных систем, то YOLOv11n лучше подходит для промышленных приложений, таких как автоматизация производственных процессов, контроль качества продукции и управление сложными беспилотными системами.

B. *RT-DETR*

RT-DETR ("Real-Time Detection Transformer") [9] — это архитектура, основанная на механизме трансформеров, специально разработанная для задач детекции объектов в реальном времени. В отличие от традиционных сверточных моделей, RT-DETR

использует механизм внимания, который позволяет анализировать глобальный контекст изображения.

Архитектура RT-DETR включает [11] в себя несколько ключевых компонентов. Backbone, например ResNet или Swin Transformer, используется для извлечения признаков из изображения. Энкодеры и декодеры трансформеров анализируют эти признаки, определяя расположение объектов и их классы.

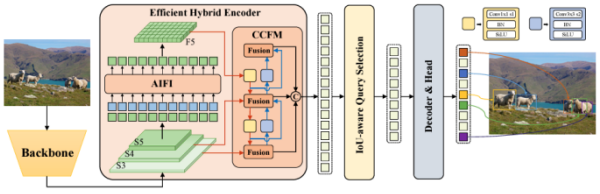


Рис. 5 Архитектура сети RT-DETR

Одной из отличительных особенностей RT-DETR является использование фиксированного количества запросов (queries), каждый из которых отвечает за предсказание одной ограничивающей рамки и её класса. Это позволяет отказаться от традиционного подавления немаксимумов (Non-Maximum Suppression, NMS), что ускоряет процесс предсказания и снижает вероятность ошибок.

C. Метрики оценки качества

Для оценки качества моделей детекции объектов широко используются метрики [12] Precision, Recall, mAP50, mAP50-95 и IoU. Каждая из них позволяет оценить различные аспекты производительности моделей.

Precision — это метрика, которая показывает долю правильно определённых объектов среди всех предсказанных моделью объектов. Высокое значение Precision свидетельствует о том, что модель редко делает ошибки в виде ложных срабатываний.

Recall — это метрика, измеряющая способность модели находить все объекты на изображении. Высокое значение Recall указывает на то, что модель эффективно обнаруживает объекты, даже если они сложноразличимы.

mAP50 (средняя точность при IoU ≥ 50%) показывает, насколько точно модель распознаёт объекты, соответствующие эталонным рамкам, с перекрытием не менее 50%. Эта метрика позволяет оценить, насколько хорошо модель локализует объекты, не требуя слишком высокой точности.

mAP50-95 — это обобщённая версия mAP50, которая учитывает диапазон значений IoU от 50% до 95% с шагом 5%. Она вычисляется как среднее значение mAP для всех значений IoU в заданном диапазоне: где — значения IoU.

IoU (Intersection over Union, пересечение-объединение) измеряет степень совпадения предсказанных рамок с эталонными и определяется формулой: где — площадь пересечения предсказанной и эталонной рамок, а — их объединённая площадь.

Метрики Precision, Recall и mAP используются для оценки общей эффективности моделей детекции, в то время как IoU даёт представление о точности локализации объектов. Совокупное использование этих метрик позволяет всесторонне оценить производительность моделей и определить их пригодность для различных приложений.

III. Сравнение

A. Метрики качества

Для лучшего понимания различий и сильных сторон YOLOv8n, YOLO11n и RT-DETR приведём их сравнительную характеристику. График, отображающий отношение train/box_loss и val/box_loss, помогает выявить разницу между ошибкой модели на обучающем наборе данных (train/box_loss) и валидационном наборе данных (val/box_loss) для моделей YOLO.

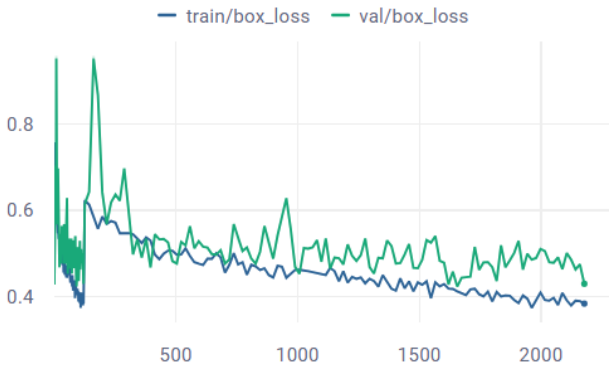


Рис. 6 Процесс обучения сети YOLOv8n

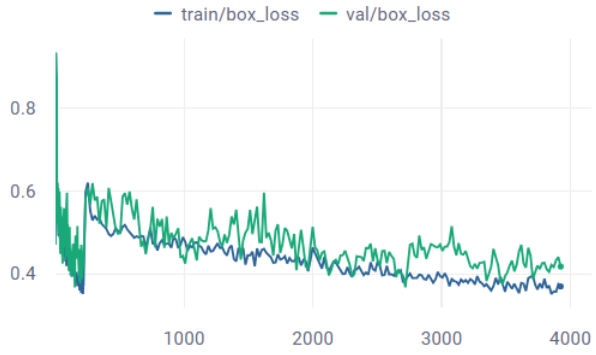


Рис. 7 Процесс обучения сети YOLOv11n

Далее можно рассмотреть, как интересующие нас метрики показывают себя на каждой из сети. Чтобы легче было воспроизвести информацию из графиков, показываем только **mAP50-95** и **Recall.**

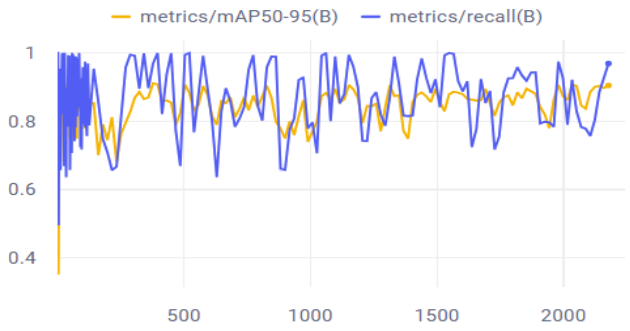


Рис. 8 Метрики обучения сети YOLOv8n

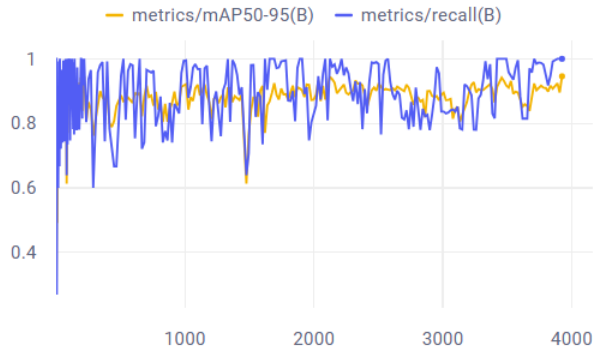


Рис. 9 Метрики обучения сети YOLOv11n

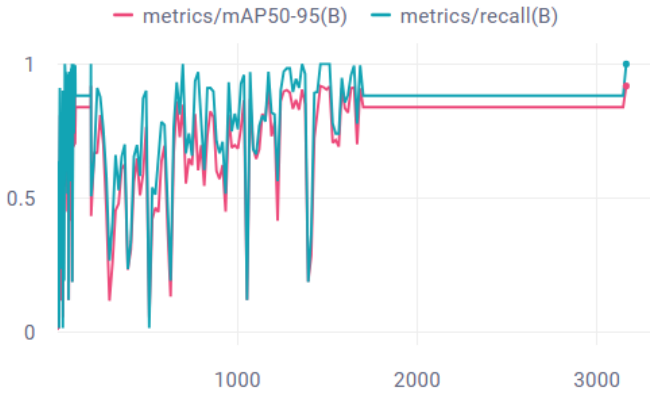


Рис. 10 Метрики обучения сети RT-DETR

Результаты всего обучения можно обобщить в следующей таблице:

Модель	Precision	Recall	mAP50	mAP50-95	Скорость обработки
YOLOv8n	0.926	1.0	0.995	0.995	60.8 ms
YOLO11n	0.985	1.0	0.995	0.995	57.0 ms
RT-DETR	0.950	1.0	0.995	0.918	42.2 ms

Табл. 1 Метрики качества моделей

Таблица демонстрирует, что результаты всех моделей относительно похожи друг на друга. YOLO11n показывает наилучшие результаты по Precision и mAP50-95, однако, учитывая заметно более высокую скорость обработки RT-DETR, выбор в пользу этой модели может быть оправданным, особенно для приложений, требующих быстрого действия.

B. Результаты детекций

Теперь посмотрим примеры обработки изображения каждой модели на тестовых данных. Приведём их визуализацию в виде сравнительных изображений. Эти примеры помогут выделить поведение модели в различных сценариях.

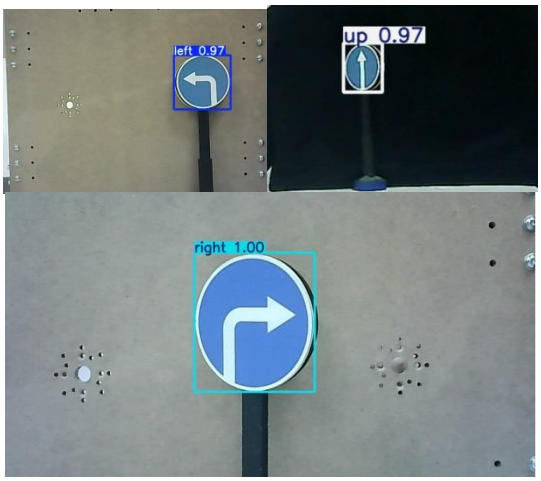


Рис. 11. Примеры обработки изображения модели YOLOv8n

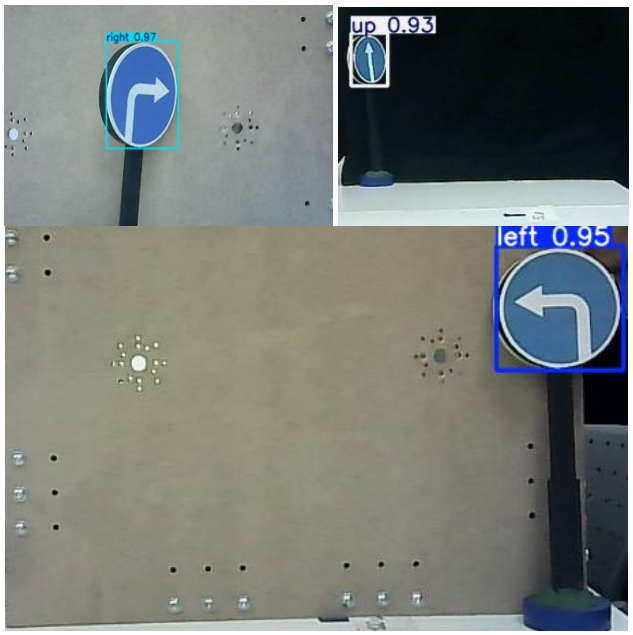


Рис. 12. Примеры обработки изображения модели YOLOv11n

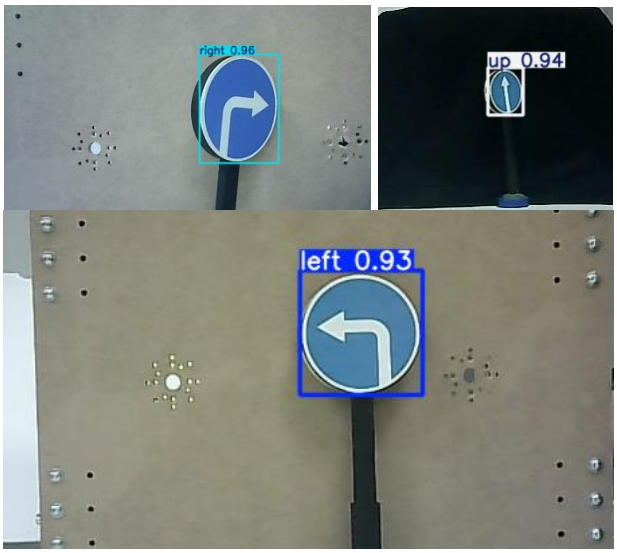


Рис. 13. Примеры обработки изображения модели RT-DETR

IV. Заключение

В данной работе рассматривалась задача классификации дорожных знаков на мобильном роботе в условиях смоделированной городской среды. Для решения задачи был использован специально собранный датасет, включающий изображения дорожных знаков, созданные с помощью 3D-модели.

В ходе исследования были протестированы три модели: YOLOv8n, YOLOv11n и RT-DETR. Все модели продемонстрировали хорошие результаты по основным метрикам оценки качества классификации, включая Precision, Recall, mAP50 и mAP50-95.

YOLOv11n обеспечила наибольшую точность за счёт применения новых компонентов и механизмов внимания, что делает её предпочтительным выбором для задач, требующих высокой точности распознавания. Модель RT-DETR, в свою очередь, продемонстрировала наивысшую скорость обработки изображений (42,2 мс на кадр), что делает её особенно подходящей для приложений с жесткими ограничениями по времени отклика.

Литература

[1] Sünderhauf, Niko, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft et al. "The limits and potentials of deep learning for robotics." The International journal of robotics research 37, no. 4-5 (2018).

[2] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," arXiv preprint arXiv:2302.12345, 2023, pp. 1-15.

[3] В. Л. Лим. Исследование вопроса распознавания Светофоров (2023) , сборник статей на тему «Искусственный Интеллект в Промышленных, Коммерческих, Медицинских и Финансовых Приложениях».

[4] Gaudenz Boesch. YOLOv8: A Complete Guide. Available at: https://viso.ai/deep-learning/yolov8-guide/. (Accessed: 29.11.2024).

[5] S. B. Berkovich, Kotov N.I., Lychagov A.S., Sadekov R.N., Sholokhov A.V., Panokin N.V, "Vision System as an Aid to Car Navigation," Proceedings of the IEEE International Conference on Vehicular Technology, 2017, pp. 2

[6] Sohan, M.; Ram, T.S.; Reddy, C.V.R. A review on YOLOv8 and its advancements. In Proceedings of the International Conference on Data Intelligence and Cognitive Informatics 2024, Tirunelveli, India, 18–20 November 2024; Springer: Singapore, 2024; pp. 529–545

[7] Khanam, Rahima, and Muhammad Hussain. "Yolov11: An overview of the key architectural enhancements." arXiv preprint arXiv:2410.17725 (2024).

[8] S Nikhileswara Rao. YOLOv11 Architecture Explained: Next-Level Object Detection with Enhanced Speed and Accuracy. Available at: https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71 (Accessed: 09.12.2024).

[9] A computer vision system for navigation of ground vehicles: Hardware and software / A. G. Bukin, R. N. Sadekov, A. S. Lychagov, O. A. Slavin // Gyroscopy and Navigation. – 2016. – Vol. 7, No. 1. – P. 66-71. – DOI 10.1134/S207510871601003X. – EDN WPUSEP.

[10] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, Jie Chen. DETRs Beat YOLOs on Real-time Object Detection (2024).

[11] Sovit Ranjan Rath. RT-DETR: Paper Explanation and Inference. Available at: https://debuggercafe.com/rt-detr/ (Accessed: 15.12.2024)

[12] В. О. Кирвяков. Исследование возможности детектирования дорожных знаков, сборник статей на тему «Искусственный Интеллект в Промышленных, Коммерческих, Медицинских и Финансовых Приложениях» (2023).

ГРНТИ: 28.23.37
ASJC: 1702
OECD: 01.02.EP

Использование нейронных сетей для определения сгенерированных изображений

С. О. Иванов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2010605@edu.misis.ru

***Аннотация*— в данной статье представлен подход к созданию нейронной сети, способной классифицировать изображения на основе их происхождения: определять, являются ли они произведением, созданным художником или результатом работы генеративных алгоритмов. Описаны архитектура модели, процесс обучения, а также проведён анализ её точности и области применения.**

Ключевые слова — искусственный интеллект, нейронные сети, анализ изображений, классификация изображений, CNN, AI Art.

I. Введение

С развитием технологий машинного обучения и генеративных алгоритмов создание визуального контента, ранее требовавшее значительных творческих усилий, стало доступным и автоматизированным процессом. Генеративные нейронные сети, такие как GAN (Generative Adversarial Networks)[1] и диффузионные модели, могут создавать изображения, которые визуально неотличимы от работ, выполненных художниками. По мере своего распространения генеративный ИИ переворачивает устоявшиеся основы креативной индустрии и выдвигает на первый план такие этические вопросы, как нарушение авторских прав, защита конфиденциальных данных и увольнение работников[2].

Хотя самые ранние попытки создания изображений с использованием ИИ датируются еще 1970-ми годами, на протяжении десятилетий в этой области не наблюдалось существенного прогресса. С появлением первых сверточных генеративно-состязательных нейронных сетей (GAN) в 2014 году началось быстрое развитие технологий в области генерации изображений, создаваемых ИИ[3]. Уже в 2015 году были впервые описаны модели диффузии [4], которые легли в основу моделей генерации изображений, таких как DALL-E, в 2020-х годах.

Рис. 1. Пример изображения, созданного с помощью VQGAN-CLIP

Во время бума ИИ в 2022 году модели преобразования текста в изображение, такие как Midjourney, DALL-E, Stable Diffusion и позднее, в 2024 году, FLUX, стали широко доступны для публики, что позволило не только художникам быстро создавать изображения с небольшими усилиями[5].

Рис. 2. Пример изображения, созданного с помощью Flux 1.1 Pro

Современные технологии искусственного интеллекта достигли значительных успехов в генерации изображений, практически неотличимых от созданных человеком. Широкая доступность разнообразных инструментов и возможность точной настройки позволяют любому желающему создавать фотореалистичные или художественные изображения.

Эти технологии находят применение в искусстве, дизайне, рекламе и других областях. Однако появление изображений, созданных ИИ, вызвало проблему их идентификации. Разница между творчеством человека и искусственно сгенерированным контентом в последнее время стала гораздо менее значительна, вследствие чего часто бывает затруднительно определить происхождение изображений.

Это открывает путь для мошенничества, подделки произведений искусства, манипуляций аудиторией, создания фальшивых фотографий людей, анализ подлинности которых подробно рассмотрен в статье[6], выдачи сгенерированных изображений за оригинальные работы художников. Кроме того, сгенерированные изображения могут часто использоваться в различных областях, от рекламы и маркетинга до сферы развлечения. Публикация или монетизация контента без указания их искусственного происхождения может вводить пользователей в заблуждение. Еще одной серьезной проблемой является нарушение авторских прав, когда нейронные сети, обученные на работах художников, воспроизводят эле-

менты их стиля, что приводит к спорам об этичности и законности подобных действий.

Тогда как в различных статьях ранее в основном рассматривался анализ изображения, определялся жанр, стиль, объекты произведения и прочие классификаторы, например в статье[7], в текущей работе будет рассматриваться только источник происхождения изображения, без определения прочих характеристик, которые могут быть слишком разнообразны и мешать поставленной задаче.

В данной статье рассматривается архитектура нейронной сети, предназначенная для решения задачи классификации изображений, направленной на определение их происхождения — нарисованы ли они художниками, или они были сгенерированы с использованием генеративных нейросетей. Архитектура построена на основе сверточной нейронной сети (CNN), включающей в себя последовательность сверточных и пулинговых слоев, предназначенных для извлечения высокоуровневых признаков из изображений размером 512x512 пикселей.

II. Наборы данных

Для обучения и дальнейшего тестирования рассматриваемой в данной работе нейронной сети, способной определять сгенерированные изображения, был собран обширный датасет из сгенерированных и нарисованных художниками изображений разного формата.

В первый набор данных, собранный на открытой площадке по генерации и публикации изображений вошли самые различные изображения, собранные в разном разрешении и в любых жанрах (Рисунок 3).

- набор данных содержит 5350 изображений, каждое из которых является сгенерированным изображением с использованием модели FLUX;
- все изображения в этом наборе размечены и классифицированы как сгенерированные.

Рис. 3. Примеры сгенерированных изображений

Во второй набор данных, собранный на открытой площадке публикации изображений, вошли нарисованные вручную художниками изображения различных форматов и размеров (Рисунок 4).

- набор данных содержит около 2500 изображений, каждое из которых является нарисованной вручную картиной;
- все изображения в этом наборе размечены и классифицированы как нарисованные вручную.

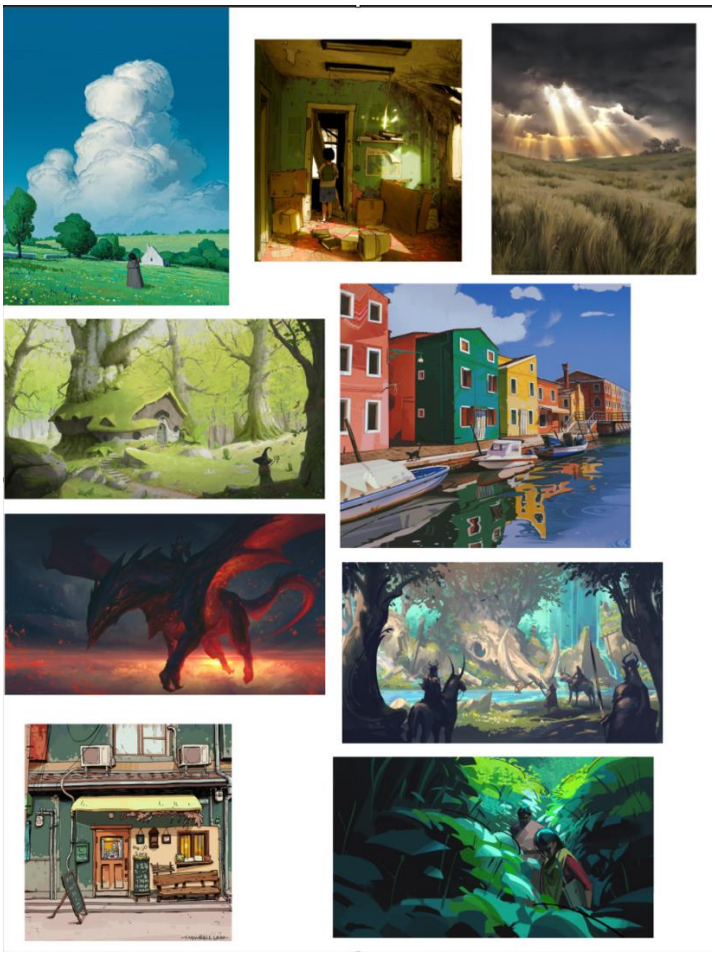


Рис. 4. Примеры нарисованных изображений

III. Нейросетевые архитектуры

Модель нейросетевой архитектуры, используемая в данной работе, представляет собой типичную сверточную нейронную сеть (CNN), оптимизированную для задачи бинарной классификации изображений на две категории – сгенерированное и нарисованное. Основные компоненты модели и их функции подробно описаны далее.

На первом этапе выполняется предзагрузка всех изображений из отобранного датасета и для обучения модели нейронной сети. Каждое изображение сжимается и обрезается до размера 512x512 пикселей и организуется классификация на две категории. Для подготовки данных используется нормализация, приводящая значения пикселей в диапазон от 0 до 1, что ускоряет обуче-

ние модели. Также в случайном порядке датасет разбирается на тренировочные данные и валидационные, чтобы модель могла обучаться с проверкой.

На втором этапе в модель вносятся сверточные слои, которые формируют основу модели. Они выполняют роль автоматического выделения признаков, таких как края, текстуры, формы и более сложные структуры. Каждый сверточный слой использует активационную функцию ReLU для введения нелинейности и увеличения выразительности модели.

После каждого сверточного слоя используется слой пулинга, который необходим для уменьшения пространственного разрешения выходных данных, снижая количество параметров и делая модель более устойчивой к локальным искажениям в изображении (Рисунок 5).

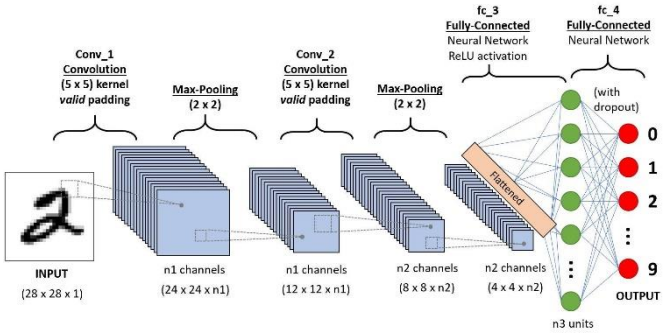


Рис. 5. Процесс обучения нейронной сети посредством свертки

После сверточных блоков используется слой Flatten, который преобразует выходные двумерные карты признаков в одномерный вектор. Этот вектор подается на полносвязный слой с 512 нейронами и функцией активации ReLU. Данный слой выполняет роль классификатора, который интегрирует признаки, выделенные предыдущими слоями[8].

Для предотвращения переобучения применяется слой Dropout, который случайным образом обнуляет 50% нейронов на этапе обучения. Это увеличивает обобщающую способность модели, особенно при работе с ограниченным набором данных

Выходной слой состоит из одного нейрона с сигмоидной функцией активации. Он возвращает значение в диапазоне от 0 до 1, где значение, близкое к 0, означает, что изображение, вероятно, нарисовано вручную. Значение, близкое к 1, означает, что изображение, вероятно, сгенерировано.

Для обучения модели используется бинарная кросс-энтропия как функция потерь, что делает её идеальной для задач бинарной классификации. Оптимизация осуществляется с помощью алгоритма Adam, который обеспечивает адаптивное обновление весов на основе градиентов, повышая скорость и эффективность обучения.

Полный список слоев представлен в таблице 1.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 510, 510, 32)	896
max_pooling2d (MaxPooling2D)	(None, 255, 255, 32)	0
conv2d_1 (Conv2D)	(None, 253, 253, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 126, 126, 64)	0
conv2d_2 (Conv2D)	(None, 124, 124, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 62, 62, 128)	0
flatten (Flatten)	(None, 492032)	0
dense (Dense)	(None, 512)	251,920,896
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 252,014,657 (961.36 MB)
Trainable params: 252,014,657 (961.36 MB)
Non-trainable params: 0 (0.00 B)

Таблица 1. Список слоев сверточной нейронной сети

IV. Тестирование и результаты

Модель оценивается по метрике точности (accuracy), что позволяет эффективно измерить её способность правильно классифицировать изображения. Для более детального анализа рассмотрим также такие метрики, как F1-score, Precision и Recall.

Для более точного обучения нейросеть обучалась на большом количестве эпох. Одна эпоха производит тренировку нейросети через весь цикл, во время которой она проходит через все входные данные. С ростом количества эпох растет точность обучения нейросети.

Для оптимального обучения нейросеть обучалась на 15 эпохах. Далее мы можем посмотреть точность на каждой эпохе, посчитать функцию потерь, чтобы доказать, что данное количество эпох оптимально для обучения нейросети.

Также построим графики:

- Recall (полноты), которая измеряет долю истинно положительных предсказаний среди всех реальных положительных примеров.

- Precision (точности), которая измеряет долю истинно положительных предсказаний среди всех предсказанных положительных.

- ROC-AUC (площадь под кривой), которая показывает способность модели различать классы на всех уровнях порога классификации:

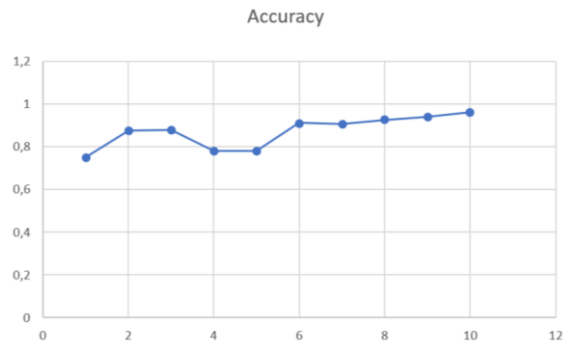


Рис. 6. Результат обучения нейросети по показателю «Accuracy»

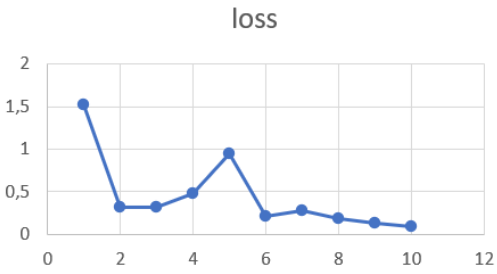


Рис. 7. Результат обучения нейросети по показателю «Функция потерь»

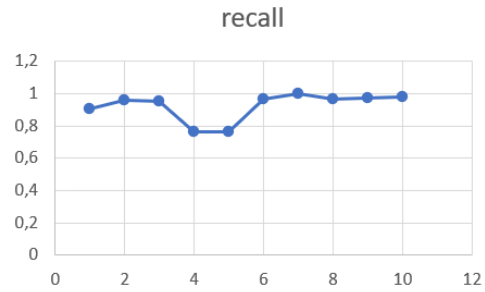


Рис. 8. Результат обучения нейросети по показателю «Recall»

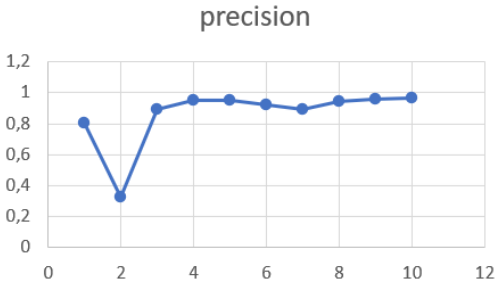


Рис. 9. Результат обучения нейросети по показателю «Precision»

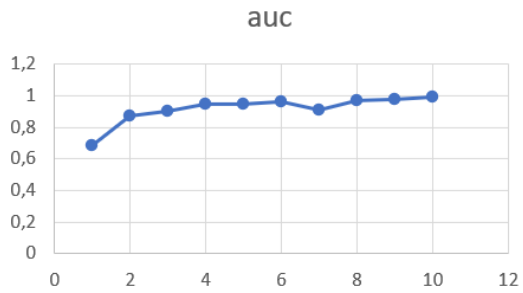


Рис. 10. Результат обучения нейросети по показателю «AUC»

В таблице 1 приведены полученные значения метрик для данного набора данных.

epoch	accuracy	val_accuracy
1	0.6167	0.8819
2	0.9062	0.8819
3	0.8386	0.8064
4	0.7812	0.8281
5	0.8484	0.9080
6	0.7812	0.9167
7	0.8931	0.9132
8	0.8125	0.9149
9	0.9259	0.9123
10	0.9062	0.9253
11	0.9083	0.8802
12	0.9875	0.8811
13	0.9475	0.8993
14	0.9912	0.8976
15	0.9757	0.9132

Таблица 2. Полученные метрики для наборов данных

Далее рассмотрим результаты тестирования полученной модели на заранее выбранной выборке изображений, не участвующих в тренировочном процессе нейронной сети. Также для тестовой части были выбраны изображения других художников, не входящих в обучающую выборку, для случая тестирования не сгенерированных изображений.

Рассмотрим полученные результаты. Из выбранных 33 изображений для теста:

6 из 8 изображений были определены правильно, как точно сгенерированные (prediction > 0.7)

23 из 36 изображений были определены правильно, как точно нарисованные человеком (prediction < 0.3)

8 изображений были ошибочно определены в неправильную категорию.

7 изображений были помечены, как неопределенные (0.3 < prediction < 0.7)

Итоговая точность модели на тестовой выборке – 81%.

Часть из выбранных для теста изображений представлена на рисунке 11. Цветом показано, правильно ли нейросеть определила происхождение изображения.

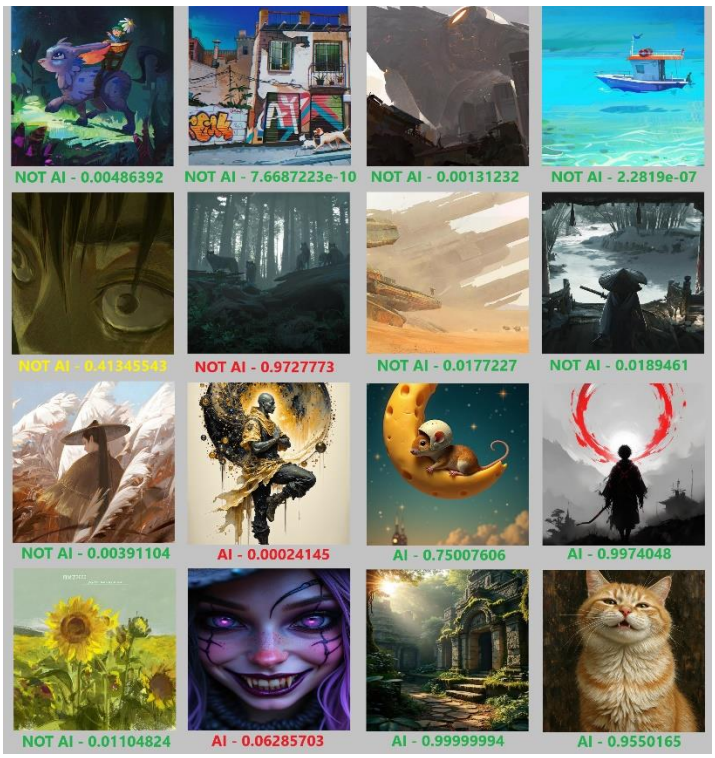


Рис. 11. Примеры изображений с определением происхождения

V. Заключение

В ходе выполнения работы была разработана и обучена нейронная сеть для классификации изображений на два класса: сгенерированные и нарисованные художником. Архитектура модели базировалась на сверточной нейронной сети (CNN), включающей последовательность сверточных, пулинговых и полносвязных слоев.

Основной метрикой для оценки выступала точность (accuracy). На тестовой выборке модель показала точ-

ность порядка 81%, что свидетельствует о ее способности различать изображения.

Разработанная модель успешно решает задачу бинарной классификации изображений с хорошей точностью и эффективностью. Дальнейшие улучшения могут быть достигнуты за счет увеличения объема обучающего набора данных и применения методов повышения устойчивости модели к визуальным искажениям в изображениях.

Литература

[1] Michał Chruściński. “A brief history of AI-powered image generation”, available at: https://sii.ua/blog/en/a-brief-history-of-ai-powered-image-generation/ (Accessed: December 28, 2024).

[2] George Lawton. “Generative AI ethics: 8 biggest concerns and risks”, available at: https://www.techtarget.com/searchenterpriseai/tip/Generative-AI-ethics-8-biggest-concerns/ (Accessed: December 28, 2024).

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", 2014, arXiv:1406.2661

[4] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, Surya Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics", 2015, arXiv:1503.03585

[5] Benj Edwards. “FLUX: This new AI image generator is eerily good at creating human hands”, available at: https://arstechnica.com/information-technology/2024/08/flux-this-new-ai-image-generator-is-eerily-good-at-creating-human-hands/ (Accessed: December 28, 2024).

[6] Злакоманов, П. Е. ИИ в детекции фэйков: Анализ подлинности лиц / П. Е. Злакоманов, И. Б. Алексеев // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 48-53. – EDN BFEAQP.

[7] Кудинов, Я. О. Исследование возможности классификации картин при помощи компьютерного зрения / Я. О. Кудинов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 106-111. – EDN QGVKPE.

[8] Ljubisa Stankovic, Danilo Mandic, " Convolutional Neural Networks Demystified: A Matched Filtering Perspective Based Tutorial", 2021, arXiv:2108.11663

ГРНТИ: 28.23.15, 06.01.29
ASJC: 1707, 2002
OECD: 01.02.EP

Распознавание ценников с целью оценки их актуальности

Р. А. Каримов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2410746@edu.misis.ru

М. Э. Насибов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2005329@edu.misis.ru

***Аннотация* — в работе рассматривается задача распознавания ценников с целью проверки их актуальности в розничной торговле с использованием нейронных сетей. Современные методы глубокого обучения, такие как YOLO и EAST, демонстрируют высокую точность в задачах распознавания текста. В статье анализируются различные подходы к оценке ценников на изображениях с использованием различных наборов данных.**

Ключевые слова — нейронные сети, распознавание текста, ценники, обработка изображений, YOLO, EAST, OCR.

I. Введение

На сегодняшний день покупатели часто сталкиваются с неправильной или устаревшей информацией на ценниках в магазинах, что может негативно сказаться на их лояльности. Это, в свою очередь, приводит к потерям как в прибыли, так и в репутации бизнеса.

В каждом магазине информация о товарах может обновляться каждые 7 дней, и в зависимости от ассортимента проверка ценников может занимать различное время, но обычно не менее нескольких часов. Сотрудники магазина часто выполняют эту задачу до открытия, однако не всегда успевают завершить её вовремя. Кроме того, проверка может проводиться и после окончания рабочего дня, что приводит к необходимости сверхурочной работы [1].

При разработке системы проверки ценников важными задачами являются обнаружение и распознавание текстовой информации [2] на ценниках и упаковках товаров. Для решения задач распознавания ценников активно применяются технологии компьютерного зрения и обработки изображений. Ключевой задачей является идентификация названия товара и его цены на изображениях, полученных с помощью камер [3]. В этом процессе используются методы, такие как сегментация и распознавание символов (OCR), которые обеспечивают точное извлечение текстовой информации. Эти подходы являются универсальными и могут быть адаптированы для различных приложений, включая распознавание ценников и другие задачи анализа визуальных данных [4].

Обнаружение текста является ключевой задачей в области компьютерного зрения, имеющей множество практических применений, таких как распознавание и поиск текста [5], обработка документов [6], мгновенный перевод [7], автопилотирование [8] и медицина [9]. Благодаря стремительному развитию методов обнаружения объектов с использованием сверточных нейронных сетей (CNN) и сегментации экземпляров, достигнуты значительные успехи в обнаружении текста стандартной формы и соотношения сторон. Таким образом, распознавание ценников, как одна из задач идентификации текста на изображениях, сталкивается с вызовами, связанными с обнаружением текста произвольной формы. Эта проблема представляет собой одну из самых сложных задач в области компьютерного зрения и продолжает вызывать растущий интерес как в научных, так и в промышленных кругах [10].

В данной работе будут рассмотрены методы глубокого обучения для решения задачи распознавания информации с ценников, а также будет проведен их сравнительный анализ.

II. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались наборы открытых данных. Рассмотрим используемые наборы данных.

A. *Price-tag-detect*

Набор данных "price-tag" представляет собой набор данных, состоящий из более чем 300 изображений ценников, которые отлично подойдут для наших целей исследования актуальности ценников в розничной торговле. Он включает в себя аннотированные данные о ценниках, собранные из различных магазинов и супермаркетов, охватывающих широкий спектр товаров и категорий. Датасет содержит информацию о ценах, датах обновления ценников, а также метаданные о товарах, такие как наименование, категория, бренд и уникальный идентификатор товара.

Каждый элемент датасета включает в себя следующие атрибуты: идентификатор товара, текущая цена, предыдущая цена (если доступна), дата последнего обновления ценника, а также информация о наличии товара на складе. Данные были собраны в различных условиях, включая разные регионы и типы магазинов, что позволяет исследовать влияние местоположения и формата торговли на актуальность цен.

Однако наличие в выборке различных магазинов несёт в себе как плюсы, так и минусы. С одной стороны, различие ценников позволит обучить модель проверять актуальность ценников в любых магазинах. С другой стороны, информация на ценниках представлена разная и расположена в разных видах ценников неодинаково (рисунок 1).

Рис. 1. Примеры ценников из различных магазинов

B. *price-tags*

Набор данных price-tags содержит около 50 изображений ценников из магазина «ЛЕНТА», созданных авторами для схожего исследования. Разметка кадров содержит в себе следующую информацию:

- информация о товаре;
- штрих -код;
- цена без скидочной карты;
- цена со скидочной картой;

Все изображения сделаны вручную при посещении магазина с устройством для фотографирования (рисунок 2).

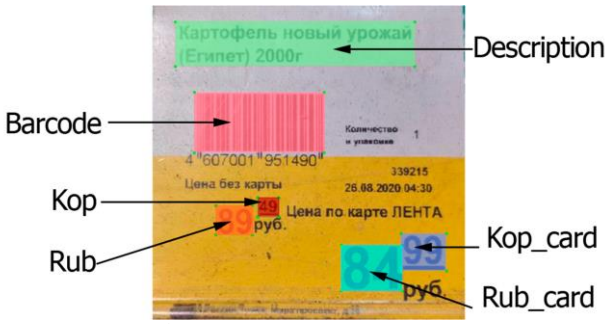


Рис. 2. Изображение с примерной разметкой

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

A. *YOLOv4-tiny+EasyOCR*

В данной работе [1] рассматривается модификация YOLOv4-tiny и использование EasyOCR для решения задачи распознавания ценников. Подход, предложенный авторами, включает в себя два этапа: детектирование и распознавание.

На первом этапе используется YOLOv4-tiny, легковесная версия популярной модели YOLO, которая предназначена для быстрого и эффективного детектирования объектов в реальном времени. Эта модель была выбрана благодаря своей высокой скорости обработки и хорошей точности, что делает её идеальной для применения в условиях, где требуется быстрое реагирование, например, в системах автоматизации торговли или в мобильных приложениях [11].

YOLOv4-tiny обучается на наборе данных, содержащем изображения ценников, что позволяет нейронной сети эффективно выделять области внутри ценника, такие как описание товара, штрих -код, цена и цена со скидкой. В процессе обучения модель оптимизируется для достижения максимальной точности детектирования, что минимизирует количество ложных срабатываний и пропусков.

На втором этапе используется EasyOCR, библиотека, основанная на глубоких нейронных сетях, для распознавания текста на выделенных областях. EasyOCR поддерживает множество языков и шрифтов, что делает её универсальным инструментом для распознавания текста. После того как YOLOv4-tiny выделяет области с ценниками, EasyOCR обрабатывает эти области и извлекает текст, что позволяет получить информацию о ценах и других характеристиках товаров.

Таким образом, предложенный подход сочетает в себе мощные возможности детектирования объектов и распознавания текста, что позволяет эффективно решать задачу автоматизации распознавания ценников. Использование YOLOv4-tiny и EasyOCR в тандеме обеспечивает высокую скорость и точность, что является критически важным для приложений, требующих обработки данных в реальном времени.

Для текущей работы представляет интерес вся модель, поскольку она выполняет именно ту работу, которую мы и хотим изучить. Данная модель была обучена для детектирования и распознавания состояния информации внутри ценника. Нейронная сеть на выходе имеет несколько классов –

«description», в котором находится описание товара внутри ценника, «barcode», в котором находится штрих-код из ценника, «price» и «price_card», в которых находятся соответственно цена и цена со скидочной картой товара.

Авторы использовали собственный датасет с ценниками из магазина «ЛЕНТА», включающий около 250 изображений. В качестве показателей эффективности были выбраны две метрики: accuracy и F1-score. Расчет данных метрик представлен формулами (1)-(4).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

Рис. 3. Пример работы системы: разметка изображений с помощью YOLOv4-tiny

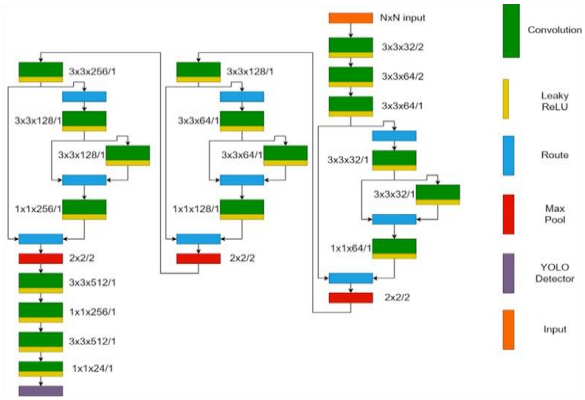


Рис. 4. Архитектура YOLOv4-tiny

Итоговая точность этой модели составляет 95,22%, что превышает точность других моделей на 20-22% [12]. При этом используемая авторами модель позволяет распознавать и впоследствии анализировать текст на 80 языках [13].

B. EAST+ResNet-BiLSTM-Attention

Другой подход заключается в использовании EAST для детекции данных на ценнике (нахождение их ограничивающих прямоугольников) и OCR-модель на основе связки ResNet-BiLSTM-Attention [3].

EAST — это метод для обнаружения текста в естественных сценах, который стремится обеспечить высокую точность и эффективность. EAST использует полностью свёрточную нейронную сеть (FCN), которая напрямую предсказывает области текста, исключая ненужные промежуточные шаги, такие как агрегация кандидатов и разбиение слов [14]. Метод может предсказывать текстовые области в виде поворотных прямоугольников (RBOX) или квадратов (QUAD), что позволяет ему работать с текстом произвольной ориентации и формы. Упрощенная структура позволяет сосредоточиться на проектировании функций потерь и архитектуры нейронной сети. EAST значительно превосходит существующие методы по точности и скорости, достигая F-меры 0.7820 на наборе данных ICDAR 2015 [15] при скорости 13.2 кадров в секунду (fps) на разрешении 720p.

TPS-ResNet-BiLSTM-Attention - это модель, используемая для распознавания текста в изображениях и сочетающая в себе несколько технологий для достижения высокой точности:

- TPS (Thin Plate Spline) - этот компонент отвечает за предварительную обработку изображений, позволяя корректировать и выравнивать текстовые строки, что улучшает качество распознавания.
- ResNet - используется для извлечения признаков из изображений. ResNet — это сверточная нейронная сеть, которая позволяет эффективно обучать глубокие модели благодаря использованию остаточных связей.
- BiLSTM (Bidirectional Long Short-Term Memory) - это рекуррентная нейронная сеть, которая обрабатывает последовательности данных в обоих направлениях (слева направо и справа налево), что позволяет лучше захватывать контекст и зависимости в тексте.
- Attention - этот механизм внимания помогает модели сосредоточиться на наиболее значимых частях входных данных, улучшая качество распознавания текста, особенно в сложных случаях, когда текст может быть искаженным или размытым.

Вместе - компоненты создают мощную архитектуру для распознавания текста, обеспечивая высокую точность и устойчивость к различным искажениям (рисунок 5).

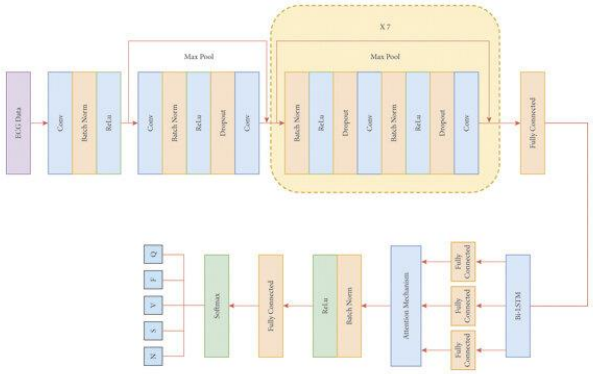


Рис. 5. Диаграмма модели TPS-ResNet-BiLSTM-Attention

Итоговая архитектура [3] EAST-ResNet-BiLSTM-Attention приведена на рисунке 6.

При проектировании архитектуры бэкенд части системы учитывалась необходимость в простоте масштабируемости и оптимизации решения [16]. Поэтому был выбран микросервисный подход, при котором каждая модель представлена в виде отдельного сервиса, а общение между сервисами происходит через очередь сообщений. Синхронизация результатов работы моделей была реализована на уровне базы данных. Такая архитектура позволяет запускать модели независимо друг от друга, что делает простым процесс эксплуатации моделей на GPU, а также оптимизацию их работы.

Авторами также реализовано облачное хранилище для хранения изображений. Результаты распознавания товарных ценников и сохраненные изображения в дальнейшем могут использоваться для дообучения системы.

Для оценки качества модели при проверке детектора использовался F-мера (f-score) и accuracy при проверке OCR-модели. Общая точность модели составляет ~90%.

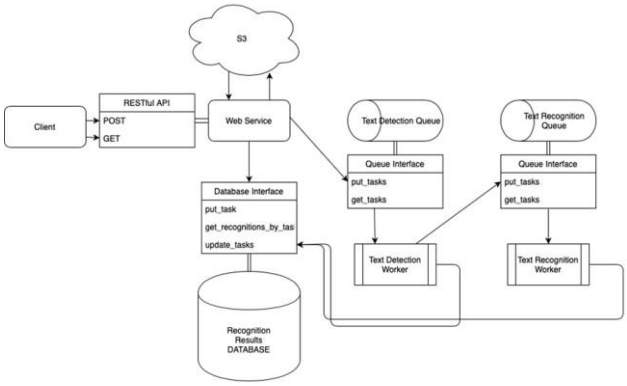


Рис. 6. Архитектура EAST+ResNet-BiLSTM-Attention

C. Комбинация математической морфологии и модели CRAFT+Tesseract

В работе [17] рассматривается использование математической морфологии и сверточных нейронных сетей для распознавания ценников.

Штрих-коды детектируются с применением математической морфологии и методов математической статистики, что позволяет повысить точность алгоритма, или с помощью сверточных нейронных сетей. Для определения цены и названия товара используется сверточная нейронная сеть CRAFT, способная обрабатывать изображения низкого качества. Найденные названия и цена нормализуются для устранения геометрических искажений и затем передаются в библиотеку Tesseract для распознавания текста. Исследование показывает, что сочетание этих методов эффективно решает задачи распознавания, учитывая требования к скорости и точности, особенно на мобильных устройствах.

CRAFT (Character Region Awareness for Text detection) предназначена для детектирования текстовых областей на изображениях. Она фокусируется на локализации отдельных символов и слов, что делает её особенно полезной для задач распознавания текста. Модель использует сверточные нейронные сети для обработки изображений. В её архитектуре применяются слои, которые позволяют эффективно извлекать признаки, необходимые для определения местоположения текста. Модель объединяет низкоуровневые и высокоуровневые признаки, что способствует более точной локализации текстовых элементов.

Одним из ключевых преимуществ CRAFT является способность обрабатывать изображения низкого качества. Это делает её особенно подходящей для распознавания текста на ценниках, где качество изображений может варьироваться. Модель генерирует два канала на выходе: один - для вероятности нахождения центра символа, а другой - для оценки связности между символами. Это позволяет точно определять, какие символы образуют слова, и группировать их соответственно.

После локализации символов с помощью CRAFT найденные названия и цены нормализуются для устранения геометрических искажений и передаются в библиотеку Tesseract для окончательного распознавания текста.

Авторами использовался собственный датасет, сделанный в местных магазинах. На первом этапе входное изображение проходило оцифровку, после над изображением выполнялись морфологические операции закрытия и эрозии, и заливались контуры (рисунок 7). В конце изображение фильтровалось с помощью математического ожидания (рисунок 8). Таким образом, выделялись области со штрих - кодами, на основе местоположения которых можно будет найти остальную информацию на изображении.

а

б

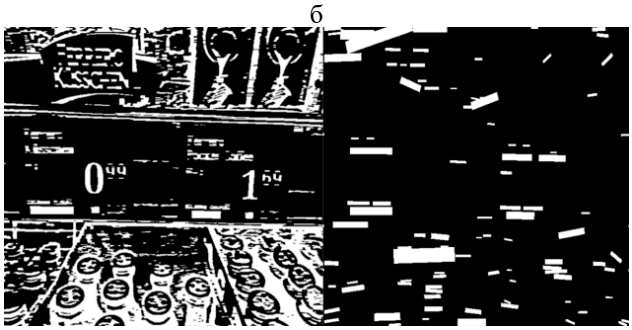


Рис. 7. а) Входное изображение (слева) и бинаризованное (справа); б) Результаты выполнения морфологических операций (слева) и заливки контуров после нее (справа)

Рис. 8. Результат фильтрации с помощью математического ожидания

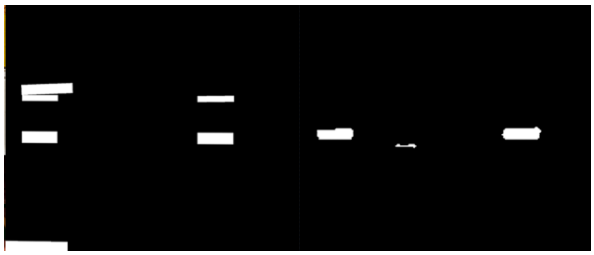
На полученных изображениях обучался детектор ~~на основе~~ модели CRAFT. Данное решение удобно для высокопроизводительных систем, однако для повышения быстродействия можно использовать математическую морфологию. Данное решение поможет использовать приложение даже на мобильных устройствах.

Авторы сравнили возможности детекции необходимых областей с помощью математической морфологии и обученной нейронной сети (рисунок 9). Методы сравнивались по 3 параметрам - precision, recall и быстродействие (таблица 1).

ТАБЛИЦА 1. Оценка детектирующей части

Метод	precision	recall	скорость, мс
Математическая морфология	0.48	0.7058	100
Обученная нейронная сеть (CRAFT)	0.9714	1.0	400

а



б в

Рис. 9. а) Входное изображение; б) Результат математической морфологии; в) Результат нейросети

Распознавание штрих - кода происходило открытой библиотекой ZXing [18]. ZXing имеет встроенные функции улучшения изображения, которые помогают декодировать штрих - код даже на плохих фото. Модель анализирует изображение попиксельно и выдает тип кода и его числовое значение. Но имеет существенный недостаток - ищет на изображении только один штрих - код, поэтому в числе задач авторов была разработка дополнительного метода для поиска всех кодов, а саму библиотеку использовать только для распознавания уже выделенных объектов.

Распознавание текста производится с помощью Tesseract OCR [19],[20] (рисунок 10). Результатом работы является преобразованное изображение с текстом в редактируемый текстовый формат. Модель использует рекуррентные нейронные сети (RNN), в частности, модификацию LSTM (Long Short-Term Memory), что позволяет ей учитывать контекст и последовательность символов при распознавании текста. После нормализации найденных названий и цен, данная модель анализирует изображения и преобразует их в текст, обеспечивая высокую точность распознавания даже на изображениях низкого качества.

Рис. 10. Результат распознавания текста с помощью Tesseract

IV. Сравнение

Сравним три описанных подхода. Для сравнения используется набор данных Price-tag-detect. Качество работы подходов складывается из качества работы детектирующей и распознающей частей. Оценка детекции производится при помощи расчёта меры Жаккара (Intersection over Union, IoU) для каждой детекции. Найденные детектором ценники с существующей разметкой с порогом 10%. Введём следующие величины:

- TP – детектор верно локализовал ценник (найдена соответствующая разметка – прямоугольники разметки и детекции пересекаются более, чем на 10%, по отношению к их общей площади).
- FP – детектор нашёл ценник там, где его нет, то есть не найдено такого прямоугольника в разметке кадра, который пересекался бы с найденным более, чем на 10%.
- FN – детектор не нашёл ценник, хотя он есть и для него есть разметка – пересечение менее, чем 10%.

Стоит отметить, что TN в данном случае не определена, так как это величина означает то, что детектор не определил ценник, где его действительно нет. По введённым величинам строятся функции оценок (1)-(3)

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Precision – сколько раз детектор нашёл ценник, где он действительно есть, по отношению к общему числу предсказанных ценников

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Recall – сколько ценников нашёл детектор из действительно присутствующих в кадрах

$$F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN} \quad (3)$$

F1 – оценка баланса между точностью (precision) и полнотой (recall). Также в случае с видеопоследовательностями, можно использовать функции MOT [22] – MOTA (multiple object tracking accuracy), которая оценивает общую точность отслеживания и детекции, и MOTP (multiple object tracking precision) – оценка точности локализации ценников (схожа с метрикой mAP [21]). Формулы (4) и (5) соответствуют формулам данных функций. При этом значения функции MOTA могут быть отрицательными – область значений $(-\infty; 1]$.

$$MOTA = 1 - \frac{FN+FP+IDS}{GT} \quad (4)$$

$$MOTP = \frac{1}{TP}\sum_i IoU_i \quad (5)$$

Здесь IoU_i – мера Жаккара i-го объекта на всей тестовой выборке. GT означает суммарное количество аннотаций, а IDS – количество потерь трека ценников. В нашем случае показатель IDS не важен, так как производится оценка именно локализации, поэтому эта величина не участвует при расчёте показателя MOTA. Таблица 2 отображает количественные оценки для трех подходов.

ТАБЛИЦА 2. Оценка детектирующей части

	YOLOv4-tiny	**EAST**	**CRAFT**
TP	294	140	231
FP	6	70	65
FN	6	96	10
Precision	0.98	0.667	0.796
Recall	0.98	0.593	0.958
F1	0.98	0.627	0.86
MOTA	0.96	0.457	0.754
MOTP	0.76	0.51	0.63

Как видно из таблицы, детектор YOLOv4-tiny имеет значительно более высокие показатели, что означает, что он намного больше находит действительных ценников и намного меньше ошибается, детектируя не относящееся к задаче окружение.

В оценку распознавательной части включены все объекты, которые входят в множество TP локализующей части. Для оценки качества распознавания текста используются различные метрики, которые помогают определить, насколько точно и эффективно система распознает текст:

- Точность (Precision)
- Полнота (Recall)
- F1-мера
- CER (Character Error Rate) – мера, которая оценивает количество ошибок на символ. CER учитывает все типы ошибок: замены, вставки и удаления.

ТАБЛИЦА 3. Оценка распознающей части

Модель	Precision	Recall	F1	CER
ResNet-BiLSTM-Attention	0.985	0.978	0.981	0.015
EasyOCR	0.968	0.962	0.965	0.023
Tesseract	0.945	0.94	0.942	0.031

Для распознавания текста на ценниках наилучшим выбором оказалась ResNet-BiLSTM-Attention, которая показала высокую точность (98.5%), низкий уровень ошибок (CER 1.5%) и превосходную производительность даже при сложных шрифтах и искажениях. Она обеспечивает минимальное количество ошибок при работе с ценами и текстовой информацией, что критически важно для точности данных, EasyOCR хоть и немного уступает первой модели по точности (96.8%) и CER (2.3%), обладает высокой скоростью обработки, что делает её подходящей для низко производительных систем, где

допустимы незначительные погрешности. Tesseract, несмотря на более высокий CER (3.1%) и несколько меньшую точность (94.5%), простая и быстрая интеграция, а также может быть использована в случаях, где точность не является критическим фактором. Таким образом, выбор модели зависит от баланса между требованиями к качеству, скорости и простоте внедрения.

V. Заключение

Были рассмотрены основные наборы данных, на которых обучались и тестировались рассматриваемые нейронные сети. Приведены три подхода к решению исходной задачи: YOLOv4-tiny+EasyOCR, предложенная и обученная авторами работы [1], EAST+ResNet-BiLSTM-Attention [3] и комбинация математической морфологии и модели CRAFT+TesseractOCR [17]. Каждая сеть рассмотрена с точки зрения её архитектуры, процесса обучения, используемых для обучения и тестирования наборов данных.

Приведённые подходы сравнивались на наборе данных Price-tag-detect. Отдельно были оценены качество детекции и распознавания ценников. Среди исследованных методов, оптимальным является комбинация YOLOv4-tiny+EasyOCR, однако если рассматривать каждый компонент по отдельности, то можно выделить YOLOv4-tiny в качестве детектора и модель ResNet-BiLSTM-Attention в качестве компонента для распознавания информации внутри ценников. Сравнивались именно конкретные модели с конкретными весами. Для сравнения архитектур в целом нужны фиксированные наборы и процессы обучения и тестирования.

Литература

[1] Neural Network-Based Price Tag Data Analysis / P. Laptev, S. Litovkin, S. Davydenko [et al.] // Future Internet. – 2022. – Vol. 14, No. 3. – DOI 10.3390/fi14030088. – EDN HPFJGB.

[2] Антонова, Т. А. Машинное распознавание и обработка текстовых символов / Т. А. Антонова, Е. Б. Дунина // Материалы докладов 57-й международной научно-технической конференции преподавателей и студентов : В двух томах, Витебск, 18–19 апреля 2024 года. – Витебск: Витебский государственный технологический университет, 2024. – С. 321-323. – EDN AODDTE.

[3] Марков, В. В. Разработка системы компьютерного зрения для распознавания товарных ценников на основе методов глубокого обучения / В. В. Марков, А. В. Марков // Студент и научно-технический прогресс : материалы XLIV научной конференции молодых учёных, Челябинск, 01–17 апреля 2020 года. – Челябинск: Челябинский государственный университет, 2020. – С. 229-231. – EDN MUVXEH.

[4] Антонов, И. А. Распознавание текстовых CAPTCHA с помощью нейронных сетей / И. А. Антонов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 17-22. – EDN WKHXPS.

[5] Ilin, D., Novikov, D., Polevoy, D.V., & Nikolaev, D.P. (2018). Fast words boundaries localization in text fields for low quality document images. International Conference on Machine Vision.

[6] Arlazarov, V.L., Arlazarov, V.V., Bulatov, K.B., Chernov, T.S., Nikolaev, D.P., Polevoy, D., Sheshkus, A.V., Skoryukina, N.S., Slavin, O.A., & Usilin, S.A. (2022). Mobile ID Document Recognition–Coarse-to-Fine Approach. Pattern Recognition and Image Analysis, 32, 89-108.

[7] Ali, B., Sadekov, R.N., & Tsodokova, V.V. (2022). A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems. Gyroscopy and Navigation, 13, 241-252.

[8] Повышение точности детектирования навигационного ориентира на основе методов компьютерного зрения в условиях недостаточной видимости / П. Ю. Беляев, И. И. Викснин, Е. А. Неверов, И. А. Зикратов // Проектирование и обеспечение качества информационных процессов и систем : Сборник докладов Международной конференции, Санкт-Петербург, 15–17 марта 2022 года. – Санкт-Петербург: Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В.И. Ульянова (Ленина), 2022. – С. 66-69. – EDN ZQYBAM.

[9] Berdichevskaia A. Atypical lexical abbreviations identification in Russian medical texts //2022 12th International Conference on Pattern Recognition Systems (ICPRS). – IEEE, 2022. – С. 1-5.

[10] Корчевский, А. С. Исследование возможности обнаружения текста произвольной формы / А. С. Корчевский // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 122-126. – EDN CQQUOD.

[11] Abdurahman, F., Fante, K.A. & Aliy, M. Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models. BMC Bioinformatics 22, 112 (2021). https://doi.org/10.1186/s12859-021-04036-4

[12] Ramil Brick, E.; Caballero Alonso, V.; O'Brien, C.; Tong, S.; Tavernier, E.; Parekh, A.; Addlesee, A.; Lemon, O. Am I Allergic to This? Assisting Sight Impaired People in the Kitchen. In Proceedings of the 2021 International Conference on Multimodal Interaction, New York, NY, USA, 18–22 October 2021; pp. 92–102.

[13] EasyOCR. URL: https://www.jaided.ai/easyocr/ (дата обращения: 23.12.2024)

[14] Zhou X. et. al. EAST: an efficient and accurate scene text detector. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017. pp. 5551–5560.

[15] D. Karatzas et al., "ICDAR 2015 competition on Robust Reading," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 2015, pp. 1156-1160, doi: 10.1109/ICDAR.2015.7333942.

[16] Baek, J. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis / J. Baek et al. // 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, doi:10.1109/iccv.2019.00481

[17] Kovtunenko, A., Yakovleva, O., Liubchenko, V., & Yanholenko, O. (2020). RESEARCH OF THE JOINT USE OF MATHEMATICAL MORPHOLOGY AND CONVOLUTIONAL NEURAL NETWORKS FOR THE SOLUTION OF THE PRICE TAG RECOGNITION PROBLEM. Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies, (1 (3), 24–31.

[18] ZXing ("Zebra Crossing") barcode scanning library for Java, Android. Available at: https://github.com/zxing/zxing (accessed 25.12.2024)

[19] Hochreiter S., Schmidhuber J. Long short-term memory. Neural computation. 1997. Т. 9, № 8. P. 1735–1780.

[20] Understanding LSTM Networks. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs (дата обращения: 25.12.2024).

[21] M. Everingham, L. Van Gool, Williams, C.K.I. et al. "The PASCAL Visual Object Classes (VOC) Challenge", International Journal of Computer Vision, 2010, vol. 88, pp. 303–338.

[22] J. Luiten, A. Ošep, P. Dendorfer et al. "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking", International Journal of Computer Vision, 2021, vol. 129, pp. 548–578.

[23] Z. C. Lipton, C. P.Elkan, B. Narayanaswamy. "Thresholding Classifiers to Maximize F1 Score", 2014 arXiv: Machine Learning, pp. 1-16.

[24] M. Sokolova, N. Japkowicz, S. Szpakowicz. "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation", Proceedings of Australasian joint conferenceon artificial intelligence, 2006, vol. 4304, pp. 1015-1021.

ГРНТИ 28.23.15
ASJC 1707
OECD 01.02.EP

Нейросетевое распознавание и мониторинг состояния водоемов на спутниковых изображениях

А. А. Катызина
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2000743@edu.misis.ru

А. Т. Фам
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2004149@edu.misis.ru

Аннотация **- в данной статье рассматривается актуальная задача нейросетевого распознавания и мониторинга состояния водоемов на спутниковых изображениях. Для её решения предлагается разработка и обучение нейронной сети, способной сегментировать водные объекты и определять их состояние на основе спутниковых снимков. В работе используются данные спутников Sentinel-1 из датасета SEN12FLOOD, а также собственная разметка, созданная с помощью инструмента CVAT. Для реализации сегментации водоемов и зон наводнений применяются нейронные сети архитектур U-Net и MobileNetV2, адаптированные для работы с мультиспектральными и SAR-данными.**

Ключевые слова — распознавание изображений с помощью нейросетей, наблюдение за состоянием водоемов, сегментация изображений, спутниковые данные, U-Net, MobileNetV2, Sentinel-1.

I. Введение

Вода является источником жизни человечества, и именно поэтому на протяжении тысячи лет водные ресурсы активно используются для поддержания жизни и промышленности. Доля пресной воды на поверхности Земли составляет всего 2,5%, из которых только 1% доступен для использования. Поэтому озера, реки, моря являются одним из наиболее важных водных ресурсов. Они используются в качестве источника водоснабжения для потребления человеком и в целом составляют около 0,3% от общего количества поверхностных источников воды [1].

Засухи и наводнения являются одними из главных проблем для устойчивого развития, поскольку они вызывают значительные социальные, экономические и экологические последствия, затрудняя доступ к ресурсам и усугубляя неравенство в пострадавших регионах. Засухи приводят к нехватке воды, снижению урожайности, голоду и экономическим потерям. Наводнения вызывают разрушение инфраструктуры, гибель людей, уничтожение сельскохозяйственных угодий и загрязнение водных источников. Например, исчезновение нескольких древних цивилизаций связывают с возникновением этих природных катастроф [2]. Также в последние десятилетия во многих регионах мира наблюдается резкий рост числа жертв и экономических потерь, вызванных засухами и наводнениями [3].

Начиная с середины XX века разрабатываются стратегии для прогнозирования наводнений и засух. Для оценки риска таких природных явлений строятся модели, которые используют уровни паводков и пиковые сбросы воды [4]. Наиболее известные модели − HEC-RAS, CCHE2D, MIKE21 и SOBEK [5-7]. Они рассчитывают движение воды по заданной территории с учетом рельефа, параметров русла реки и гидрологических данных, таких как осадки и расход воды. Модель может учитывать разные сценарии, включая изменение уровня воды, плотины или наводнения, помогая предсказывать возможные затопления. Однако такие подходы имеют ряд недостатков: сложность моделирования для урбанизированных зон, необходимость большого количества параметров, сложности с разнообразием климата.

В отличие от наводнений засуха является долгосрочным эффектом. Два наиболее распространенных подхода для раннего выявления засухи — это динамическое и статистическое моделирование. Первое - включает методы регрессионного анализа, вероятностные и стохастические модели, а также подходы на основе искусственного интеллекта [8,9]. Динамическое моделирование - с использованием данных в реальном времени, таких как осадки, речной сток, температура и результаты дистанционного зондирования, позволяет создавать системы для мониторинга и прогнозирования засух [10-12]. Такой подход требует значительных вычислительных ресурсов и сложных моделей, что затрудняет его применение отдельными пользователями. Статистические модели, основанные на регрессиях, проще, но их линейные предположения ограничивают точность долгосрочных прогнозов. Вероятностные модели обеспечивают лучшую производительность, но требуют больше вычислительных ресурсов.

С развитием искусственного интеллекта появилась возможность сократить участие человека во многих аспектах жизни. Компьютерное зрение позволило машинам интерпретировать и понимать визуальную информацию, что создало обширные перспективы для автоматизации процессов, связанных с анализом изображений и видео [13,14]. В последних исследованиях было представлено множество методов мониторинга с использованием нейронных сетей [4,9,15,16], такие как ANN, ANFIS и SVM или простые нейронные сети с несколькими слоями, которые демонстрируют лучшие результаты, особенно для долгосрочных прогнозов [17]. Данные методы не только улучшили точность оценки, но и минимизировали участие человека, а также автоматизировали процесс прогнозирования.

Эволюция подходов в области нейронных сетей привела к появлению моделей, ориентированных на решение более сложных задач, таких как распознавание объектов на спутниковых снимках и их классификацию [18,19]. Свою популярность приобрели сети U-Net, разработанные в 2015 году, изначально использующиеся для сегментации медицинских изображений [20,21]. Для выделения затопленных территорий авторы [22, 23, 24] объединяют классическую U-Net с механизмами внимания, где в качестве энкодеров используются известные предобученные модели, как MobileNetV2, EfficientNet-B7.

До сих пор сети U-Net активно применялись только для задачи сегментации изображений. Однако в данной работе будет рассмотрено применение модифицированной сети U-Net для задачи классификации спутниковых изображений на наличие и отсутствие наводнений. Кроме того, будет проанализирована целесообразность использования трансферного обучения для данной задачи. В качестве предобученной модели выбрана MobileNetV2 [25], так как она показала высокую точность в задачах классификации объектов на спутниковых снимках, изображениях дистанционного зондирования и космических снимках [26,27,28].

II. Наборы данных

Для проведения исследования и обучения нейронной сети использовались данные из датасета SEN12FLOOD, включающего более 3000 спутниковых снимков, полученных с использованием данных Sentinel, что позволяет учитывать как SAR (радарные) данные, так и мультиспектральные изображения [29]. Этот набор данных включает регионы, подверженные наводнениям, в различных частях мира и охватывает широкую временную шкалу. Изображения были выбраны таким образом, чтобы отразить разнообразные погодные условия, времена года и типы ландшафта, что позволяет создавать модели с высокой обобщающей способностью для детектирования наводнений в различных условиях.

Датасет содержит изображения как затопленных территорий, так и регионов, не подвергшихся наводнениям, что позволяет модели обучаться на сбалансированной выборке и корректно распознавать как присутствие, так и отсутствие наводнения. Снимки SAR от Sentinel-1 позволяют фиксировать изменения поверхности даже при облачности и сложных погодных условиях. SAR-данные полезны для обнаружения воды на затопленных территориях, поскольку они не зависят от освещенности и погодных условий [30].

Каждое изображение представлено в формате TIFF (Tagged Image File Format), что позволяет сохранить все необходимые спектральные данные. Размер изображений составляет 256 x 256 пикселей для упрощения обработки и снижения вычислительных затрат при обучении нейронной сети.

Для создания точных масок зон наводнений использовался инструмент разметки CVAT (Computer Vision Annotation Tool) [31]. В процессе аннотирования изображения размечались вручную, выделяя затопленные участки с использованием полигональных масок.

Изображения, содержащие более одного спектрального канала, были преобразованы в трехканальный формат (RGB) для загрузки в CVAT. Это позволило обрабатывать данные в совместимом с инструментом формате, сохранив при этом все необходимые детали.

Полигональная разметка зон наводнений обеспечивает высокую точность в определении границ водоемов и наводнений, что особенно важно для обучения нейронных сетей в задачах сегментации.

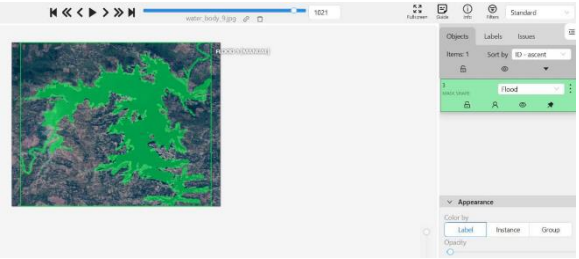


Рисунок 1 - Разметка датасета в CVAT.ai

Полученный набор данных стал основой для обучения нейронной сети. Датасет был разделен на тренировочную (80%), валидационную (20%) выборки. Такой подход обеспечил модели доступ к разнообразным погодным условиям, временным интервалам и ситуациям как с наводнениями, так и без них, что способствовало улучшению её обобщающей способности.

III. Нейросетевая архитектура U-Net

Основной идеей U-Net является построение симметричной архитектуры типа «энкодер-декодер», где нисходящая часть (encoder) извлекает признаки и сжимает пространственную размерность, а восходящая часть (decoder) восстанавливает пространственное разрешение, комбинируя его с высокоуровневыми признаками с помощью skip connections [20].

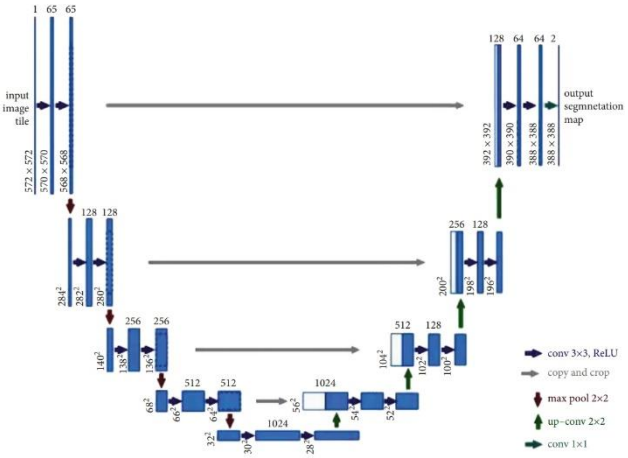


Рисунок 2. U-Net архитектура

Сеть U-Net можно разделить на три части: нисходящая часть (Encoder) – для извлечения признаков, что реализуется с помощью сверток и пулинга, бутылочное горлышко (BottleNeck) – узкая часть сети, тут происходит извлечение наиболее обобщенных признаков, и восходящая часть (Decoder) – для восстановления пространственного разрешения изображения с использованием upsampling.

Свойства симметричности, skip connections и гибкости делают из U-Net мощную архитектуру для задач сегментации [21]. В нашей работе будет использована модифицированная версия U-Net для задачи классификации, где U-Net используется как механизм извлечения признаков. Выходы ветки U-Net будут подаваться на вход в слои Flatten и Dense для выполнения бинарной классификации изображения, то есть добавляется классификационная голова [22].

На рисунке 3 представлен пайплайн метода. Изображения переводятся из TIFF формата в JPG, после чего вручную размечаются области затопления и водные участки, далее разметка сохраняется в XML формате и подается на вход модели, после она выдает предсказанный класс.

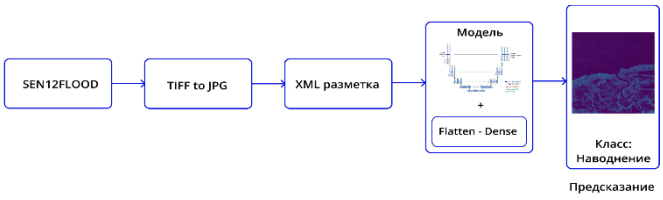


Рисунок 3. Пайплайн метода

IV. НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА MobileNetV2

MobileNetV2 представляет собой усовершенствованную версию нейронной сети MobileNet, разработанную в 2018 году для выполнения задач классификации, сегментации и анализа изображений [25]. Её основная цель — обеспечить высокую производительность при минимальных вычислительных затратах.

Ключевыми особенностями MobileNetV2 являются инвертированные остаточные связи (Inverted Residual Connections) и линейные узкие места (Linear Bottlenecks) [26]. Эти механизмы выполняют расширение признаков на входе блока, их обработку, а затем возвращают к суженному представлению. Такой подход минимизирует количество параметров без потери точности.

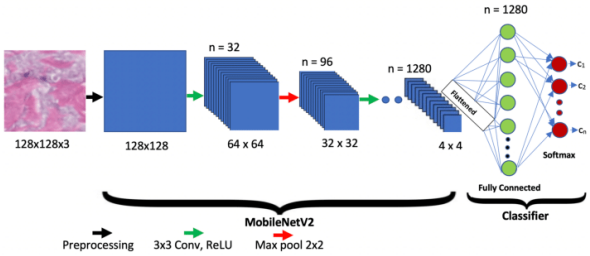


Рисунок 4. MobileNetV2 архитектура

Линейные узкие места играют важную роль в предотвращении потери мелких признаков, часто исчезающих из-за использования функции активации ReLU в стандартных архитектурах. MobileNetV2 заменяет эту активацию линейным преобразованием, что особенно эффективно для задач анализа текстур и контуров [27].

Архитектура также включает глубинно-разделяемые свёртки (Depthwise Separable Convolutions), которые значительно снижают вычислительную сложность, сохраняя производительность (рисунок 4). Настраиваемые параметры, такие как множители ширины и разрешения, делают модель гибкой и адаптивной к различным условиям использования [28].

MobileNetV2 показывает отличные результаты в задачах анализа спутниковых изображений. Модель точно определяет объекты, выделяет зоны затоплений и границы водоёмов, оставаясь при этом ресурсосберегающей.

V. ОЦЕНКА ТОЧНОСТИ

Для оценки качества работы модели использовались общеизвестные и распространенные метрики.

Обозначим, TP (True Positive) — верно предсказанные изображения, относящиеся к положительному классу (наводнение есть, и оно верно определено), FP (False Positive) — изображения, ошибочно отнесённые к положительному классу (наводнение отсутствует, но модель его определила), TN (True Negative) — верно отнесённые к отрицательному классу (наводнение отсутствует, и это правильно определено), FN (False Negative) — изображения, которые ошибочно отнесены к отрицательному классу (наводнение есть, но модель его не обнаружила). Для обучения обеих моделей U-Net и MobileNetV2 использовался learning_rate = 0.001, оптимизация Adam, функция потерь – categorical_crossentropy, так как задача является бинарной классификацией.

В таблице 1 и 2 представлены результаты работы моделей.

Таблица 1 - Метрики U-Net на валидационной выборке

Метрика	Значение
F1 - score	0.8117
Precision	0.8863
Accuracy	0.8663

Таблица 2 - Метрики MobileNetV2 на валидационной выборке

Метрика	Значение
F1 - score	0.8030
Precision	0.7690
Accuracy	0.8555

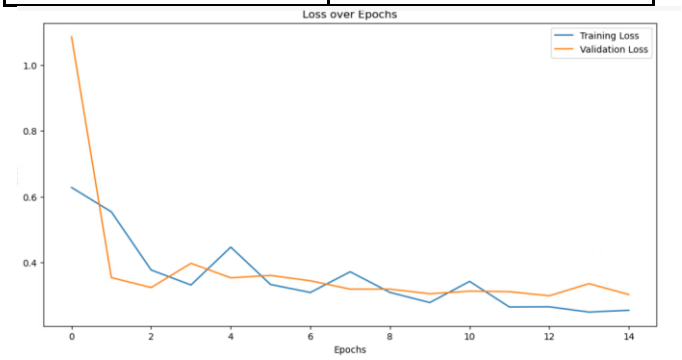


Рисунок 5 - График функции потерь для U-Net во время обучения. Для обучения модели понадобилось 12 эпох, потери при обучении составили меньше 0.4 на тренировочной и валидационной выборках.

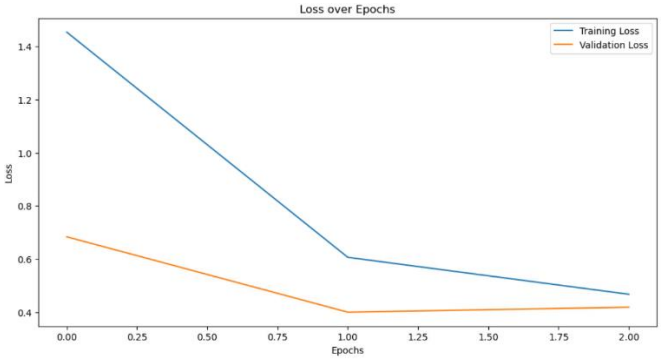


Рисунок 6 – График функции потерь для MobileNetV2. Модель обладает быстрой сходимостью, для обучения потребовалось 2 эпохи.

Полученные результаты показывают, что предложенная модифицированная архитектура U-Net способна достигать высокой точности для задачи классификации затопленных областей на спутниковых снимках.

Сравнение моделей показывает, что U-Net превосходит MobileNetV2 по значениям F1-score, Precision и Accuracy. U-Net предпочтительнее использовать для задач, где важна точность, в то время как MobileNetV2, из-за своей меньшей вычислительной сложности и более быстрой сходимости подходит для случаев, где требуется оперативность и вычислительные ресурсы ограничены.

VI. Заключение

В ходе работы был рассмотрен метод нейросетевого распознавания и мониторинга состояния водоёмов на спутниковых изображениях. Разработанные нейронные сети, построенные на основе архитектур U-Net и MobileNetV2, продемонстрировали высокую эффективность в задачах классификации и сегментации водных объектов. U-Net позволила с высокой точностью выделять контуры водоёмов и зоны наводнений, что автоматизировало процесс мониторинга и ускорило анализ состояния водных ресурсов. MobileNetV2, благодаря своей компактной и оптимизированной структуре, обеспечила эффективную классификацию с минимальными вычислительными затратами, что делает её подходящей для использования в реальном времени и на устройствах с ограниченными ресурсами.

Использование спутниковых данных Sentinel-1 в сочетании с собственноручно размеченными масками водоёмов показало, что нейросетевые подходы способны значительно повысить точность и оперативность анализа экологической обстановки, минимизируя необходимость в ручной работе и снижая временные и финансовые затраты на проведение исследований. Применение U-Net и MobileNetV2, способных учитывать особенности мультиспектральных и SAR-данных, обеспечило точное определение границ водных объектов даже в условиях облачности и сложного рельефа.

Кроме того, предложенный метод обеспечивает возможность отслеживания изменений водоёмов во времени, что особенно актуально при оценке последствий наводнений, засух и других природных явлений. В будущем планируется расширить объём данных за счёт использования дополнительных источников спутниковых снимков и улучшить качество разметки с помощью автоматизированных инструментов. Также предполагается интеграция нейронных сетей с временными рядами данных для более точного анализа динамики изменений водоёмов, что позволит не только отслеживать текущее состояние водных объектов, но и прогнозировать их поведение в будущем.

Дополнительно возможен анализ состояния воды (чистота, уровень загрязнённости) на основе мультиспектральных данных, что расширит функциональные возможности системы и повысит её ценность для экологических исследований и природоохранных мероприятий.

Литература

[1] Vasistha, P., Ganguly, R. Water quality assessment of natural lakes and its importance: An overview. Materials Today: Proceedings, p. 544–552, 2020.

[2] Munoz, S. E., Gruley, K. E., Massie, A., Fike, D. A., Schroeder, S., and Williams, J. W.: Cahokia's emergence and decline coincided with shifts of flood frequency on the Mississippi River, P. Natl. Acad Sci. USA, 112, 6319–6324, doi:10.1073/pnas.1501904112, 2015.

[3] Baldassarre, D., Montanari, G., Lins, A., Koutsoyiannis, H., Brandimarte, D., Bloeschl, L. Flood fatalities in Africa: from diagnosis to mitigation, Geophysical Research Letters, 37, doi:10.1029/2010GL045467, 2010.

[4] Yang, T.H., Liu W.C. A General Overview of the Risk-Reduction Strategies for Floods and Droughts. Sustainability 12, 2687. https://doi.org/10.3390/su12072687, 2020.

[5] DHI MIKE 21: 2D Modelling of Coast and Sea. DHI Water & Environment Pty Ltd. Hørsholm: Denmark, 2012.

[6] Shih, S.S., Kuo, P.H., Lai, J.S. A nonstructural flood prevention measure for mitigating urban inundation impacts along with river flooding effects. J. Environ. Manag. 2019, 251, 109553.

[7] Doong, D.J., Lo, W., Vojinovic, Z., Lee, W.L., Lee, S.P. Development of a new generation of flood inundation maps-A case study of the coastal city of Tainan, Taiwan 2016, 8, 521.

[8] Fung, K.F., Huang, Y.F., Koo, C.H., Soh, Y.W. Drought forecasting: A review of modeling approaches 2007–2017. J. Water Clim. Change 2019, 236.

[9] Mishra, A.K., Singh, V.P. Drought modeling: A review. Journal of Hydrology, 403, 157-175, 2011.

[10] Luo, L., Wood, E. F. Monitoring and predicting the 2007 U.S. drought. Geophysical Research Letters, 34(22), Article L22702. https://doi.org/10.1029/2007GL031673, 2007

[11] Bae, D.H., Son, K.H., Ahn, J.B., Hong, J.Y., Kim, G.S., Chung, J.S., Jung, U.S., Kim, J.K. Development of real-time drought monitoring and prediction systems in the Korea & East Asia region. Atmosphere 2012, 22, 267–277.

[12] Li, Y., Yuan, X., Zhang, H., Wang, R., Wang, C., Meng, X., Zhang, Z., Wang, S., Yang, Y., Han, B. Mechanisms and early warning of drought disasters: Experimental drought meteorology research over China. Soc. 2019, 100, 673–687

[13] Kudryashov, A.A., Mishchanin, M.A., Sadekov, R.N. Food recognition using deep learning networks and order history for smart canteen checkout automation. Smart Meal Service LLC, 2022.

[14] Guzhva, N. S., Prun, V. E., Postnikov, V. V., Lobanov, M. G., Sadekov, R. N., Sholomov D. L. Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene, 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[15] Svoboda, M., Fuchs, B. Handbook of Drought Indicators and Indices, WMO-No. 1173. World Meteorological Organization: Geneva, Switzerland, 2016.

[16] Adamowski, J., Belayneh, A. Drought Forecasting. Exploring Natural Hazards: A Case Study Approach, 207, Taylor & Francis Group: Abingdon, UK, 2018.

[17] Mokhtarzad, M., Eskandari, F., Vanjani, N.J., Arabasadi, A. Drought forecasting by ANN, ANFIS, and SVM and comparison of the models. Environ. Earth Sci. 2017, 76, 729.

[18] Грищенко, Д. И. Классификация земного покрова и землепользования / Д. И. Грищенко // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 30-35. – EDN AYJWGA.

[19] Дедов, А. Д. Обнаружение кораблей на спутниковых изображениях с использованием компьютерного зрения / А. Д. Дедов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 36-41. – EDN RVELMU.

[20] Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications, in IEEE Access, vol. 9, pp. 82031-82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[21] Yin, X.-X., Sun, L., Fu, Y., Lu, R., Zhang, Y. U-Net-Based Medical Image Segmentation // Journal of Healthcare Engineering. — 2022. — Vol. 2022. — Article ID 4189781. — DOI: 10.1155/2022/4189781

[22] Li, W., Wu, J., Chen, H., Wang, Y., Jia, Y. and G. Gui. U-Net Combined With Attention Mechanism Method for Extracting Flood Submerged Range, in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 6588-6597, 2022, doi: 10.1109/JSTARS.2022.3194375.

[23] Sakthi, Jai, S.M., Dhanya, P., Kumar, S. Detection of Flooded Regions from Satellite Images Using Modified UNET. 10.1007/978-3-030-92600-7-16, 2021.

[24] Mesvari, M., Shah-Hosseini, R. Flood Detection Based on UNet++ Segmentation Method Using Sentinel-1 Satellite Imagery. Earth Observation and Geomatics Engineering, 7(1): -. doi: 10.22059/eoge.2023.359463.1139, 2023.

[25] Mark, S., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520

[26] Wijaya, B.A., Gea, P.J., Gea, A.D., Sembiring, A., Hutagalung, C.M.S. Satellite Images Classification using MobileNet V-2 Algorithm. SinkrOn. 2023. Vol. 8, No. 4. P. 2316–2326. DOI: 10.33395/sinkron.v8i4.12949. License: CC BY-NC 4.0.

[27] Barman, T., Susan, S. Multi-Label Remote Sensing Image Classification using MobileNetV2. 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). Kamand, India: IEEE, 2024. DOI: 10.1109/ICCCNT61001.2024.10725506

[28] Jordan, J., Posada, D., Zuehlke, D., Radulovic, A., Malik, A., Henderson, T. Satellite Detection in Unresolved Space Imagery for Space Domain Awareness Using Neural Networks. AAS/AIAA Astrodynamics Specialist Conference. Charlotte, North Carolina, August 7-11, 2022. arXiv:2207.11412. DOI: https://doi.org/10.48550/arXiv.2207.11412.

[29] Clément, R., Audebert, N., Koeniguer, E., Le Saux, B., Crucianu, M., Datcu, M.. SEN12-FLOOD : a SAR and Multispectral Dataset for Flood Detection. IEEE Dataport; 2020. Available from: https://dx.doi.org/10.21227/w6xz-s898

[30] Sentinel-1 // SentiWiki URL: https://sentiwiki.copernicus.eu/web/sentinel-1 (дата обращения: 02.11.2024).

[31] Documentation // CVAT.ai URL: https://sentiwiki.copernicus.eu/web/sentinel-1 (дата обращения: 21.11.2024).

ГРНТИ 28.23.37, 73.47.12, 28.29.03
ASJC 1702
OECD 01.02.EP

Локальные методы планирования траекторий на основе обучения с подкреплением

С. Д. Киселев
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m1900033@edu.misis.ru

А. В. Алтунян
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m1801726@edu.misis.ru

***Аннотация* — в связи с необходимостью повышения безопасности в условиях растущего числа транспортных средств автоматизированные транспортные системы требуют эффективных решений для управления движением, включая выбор траектории и перестроение на многополосной дороге. Для исследования разработана симуляционная среда, моделирующая сложные сценарии дорожного движения, где агент принимает решения, взаимодействуя с динамическими объектами, находящимися на дороге. В работе рассматриваются и сравниваются два современных подхода обучения с подкреплением: Rainbow DQN, использующий механизмы приоритетного воспроизведения опыта и дуэльной архитектуры, и графовая нейронная сеть (GNN) на основе Double DQN. Особое внимание уделено анализу эффективности каждого из подходов в задачах оптимизации времени прохождения дистанции и предотвращения столкновений.**

Ключевые слова — обучение с подкреплением, Rainbow DQN, графовые нейронные сети, автоматизированные транспортные системы, принятие решений, управление движением

I. Введение

Системы автономного управления транспортными средствами являются ключевым направлением исследований в области искусственного интеллекта и робототехники [1]. Они обеспечивают возможность выполнения сложных задач, таких как выбор оптимальной траектории, предотвращение столкновений и адаптация к изменяющимся условиям на дороге. Основой этих систем являются методы глубокого обучения и обучения с подкреплением (RL), которые позволяют агентам принимать решения в условиях неопределенности [2]. Дополнительно подходы, описанные в современных исследованиях, показывают важность учета как статических, так и динамических факторов в задаче принятия решений на основе RL. Современные исследования показывают значимость гибридных подходов, объединяющих методы глубокого обучения и анализа структурированных данных, для улучшения точности и эффективности интеллектуальных систем. В частности, подходы, использующие трансформеры и графовые сети, демонстрируют потенциал в задачах обработки изображений и принятия решений, что актуально для разработки систем автономного управления [3], [4]. Это включает в себя не только адаптацию к многополосным дорожным условиям, но и возможность обучения с использованием гибридных архитектур, объединяющих графовые структуры и механизмы глубокого обучения. Также исследования показывают, что сочетание методов машинного обучения и интеллектуального управления позволяет улучшить адаптивность систем, что особенно важно для работы в сложных условиях дорожного движения [5], [6].

Одной из центральных проблем в области RL для управления движением является выбор подходящей архитектуры нейронной сети. Существуют различные классы нейросетевых моделей, которые успешно применяются в задачах прогнозирования и управления:

1. Классические полносвязные сети (MLP). Используются для обработки ограниченного объема данных. Несмотря на простоту, они неэффективны в обработке многомерной информации, характерной для дорожных сценариев [7].
2. Сверточные нейронные сети (CNN). Применяются для извлечения пространственных признаков, например, изображения с камер. Однако для задач управления, где входные данные являются структурированными, их использование ограничено [8].
3. Рекуррентные сети (RNN, LSTM, GRU). Эффективны при работе с последовательными данными, например, для предсказания траекторий объектов. Их ограничение заключается в высокой вычислительной сложности [9].
4. Графовые нейронные сети (GNN). Используются для моделирования взаимодействий между объектами, что делает их подходящими для задач, связанных с движением нескольких участников дорожного движения [10].

Среди RL-методов выделяется Rainbow DQN, который объединяет улучшения, такие как двойное Q-обучение, дуэльные сети и механизм приоритетного воспроизведения опыта [11], [12]. Эти улучшения делают его одним из самых успешных подходов в RL. В то же время графовые нейронные сети предоставляют возможность моделировать сложные зависимости между участниками движения, что снижает вычислительные затраты при развертывании.

Цель данной работы — сравнить производительность Rainbow DQN и GNN в задаче принятия решений на пятиполосной дороге с девятью динамическими объектами. Основное внимание уделено оценке качества принятых решений, их устойчивости к изменяющимся условиям и вычислительным затратам.

II. Наборы данных

Для моделирования и тестирования среды разработан процедурно генерируемый набор данных, который обеспечивает разнообразие сценариев дорожного движения и адаптируется к потребностям алгоритмов обучения с подкреплением.

Набор данных включает состояния 10 объектов: одного агента (управляемого алгоритмом RL) и девяти динамических объектов, имитирующих транспортные средства на дороге. Состояние каждого объекта представлено вектором, содержащим следующие параметры:

- x, y — координаты объекта в пространстве;
- скорость объекта (v), которая варьируется от 10 до 35 м/сек;
- полоса движения ($lane$), которой принадлежит объект;
- размерность транспортного средства ($size$).

Характеристики среды:

- Многополосная дорога. Среда состоит из пяти полос движения, каждая шириной 10 метров. Общая ширина дороги составляет 50 метров.
- Динамика объектов. Динамические транспортные средства случайным образом изменяют скорость и с вероятностью 5% выполняют перестроение между полосами.
- Перезапуск объектов. Когда динамический объект покидает видимую область агента , например, уезжает за пределы дальности обзора, он заменяется новым объектом, появляющимся в зоне видимости. Это позволяет сохранить постоянное количество взаимодействующих объектов.
- Целевая длина маршрута. Агент стремится преодолеть расстояние в 1000 метров за минимальное время.

Рассмотрим прежде всего агента. Он имеет на выбор пять действий:

- статичное ускорение (определяется средой);
- продолжать ехать вперед без изменений;
- затормаживание до 0 км/ч;
- перестроение влево;
- перестроение вправо.

Так как предполагается запуск среды чаще 30 раз в секунду, то такой набор действий нас вполне устраивает.

Для процедурного создания объектов используется алгоритмический подход. Каждый объект генерируется с начальным положением и скоростью, которые задаются случайным образом в заданных пределах. Ниже приведены ключевые моменты генерации:

- начальные координаты объекта задаются так, чтобы он располагался на определенной полосе дороги, а -координата соответствовала середине полосы;
- скорость каждого объекта изменяется случайным образом с учетом ограничения скорости и текущего состояния движения;
- перестроение между полосами происходит только если оно безопасно и возможно по правилам, встроенным в логику среды.

Разработанная среда обладает рядом характеристик, которые делают ее удобным и эффективным инструментом для обучения и тестирования алгоритмов с подкреплением.

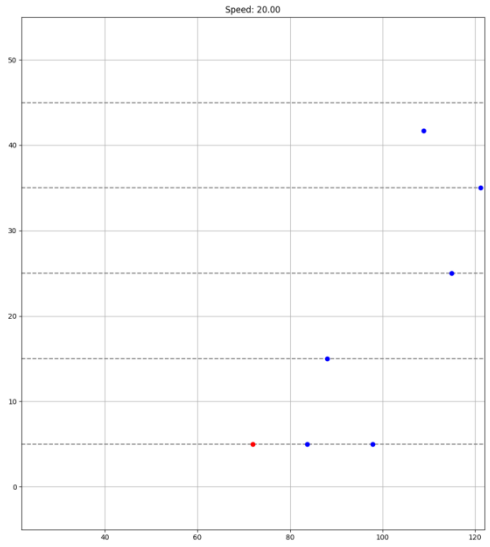


Рис. 1. Среда запуска (красная точка — агент, синяя — движущиеся препятствия)

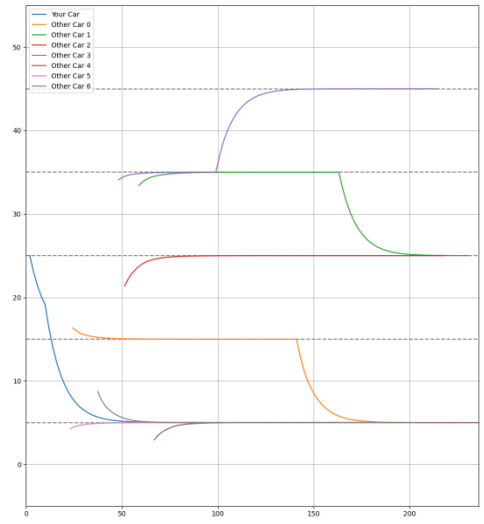


Рис. 1. Обучаемый агент (синий) и пройденные маршруты

Одним из ключевых достоинств является ее реализм: среда моделирует типичные дорожные ситуации, такие как плотное движение, перестроения транспортных средств и изменения их скоростей. Это позволяет агенту учиться принимать решения в условиях, приближенных к реальным сценариям.

За счет настройки параметров среда может быть применима к различным условиям. Код для создания и настройки среды позволяет определять характеристики объектов, динамику их поведения и взаимодействие с агентом, что дает возможность тестировать различные конфигурации и адаптировать сценарии под требования конкретного алгоритма.

Еще одним важным преимуществом является масштабируемость. Процедурная генерация объектов позво-

ляет создавать практически неограниченное количество данных, исключая необходимость использования фиксированных статических наборов. Это особенно актуально для задач обучения с подкреплением, где для достижения хороших результатов требуется большое количество данных для тренировок.

Такая структура набора данных предоставляет агенту широкий спектр ситуаций для обучения, делая среду универсальным инструментом для разработки алгоритмов управления в автоматизированных транспортных системах.

III. Нейросетевые архитектуры

Для решения задачи принятия решения по выбору действия из предложенных (ускорение, продолжать ехать вперед без изменений, торможение, перестроение влево, перестроение вправо) в настоящей работе были выбраны две нейросетевые архитектуры: Rainbow DQN и графовая нейронная сеть (GNN) на основе Double DQN. Эти подходы были выбраны в силу их доказанной эффективности в задачах обучения с подкреплением и их способности моделировать сложные взаимодействия между агентами.

A. Rainbow DQN

Rainbow DQN является улучшенной версией классического алгоритма DQN, в которой объединены несколько ключевых методов, включая двойное Q-обучение, дуэльную архитектуру, приоритетное воспроизведение опыта (PER) и использование шумовых слоев [13]. В данной работе реализована модифицированная версия сети с архитектурой ImprovedDuelingNet, а также с использованием механизма внимания и улучшенного механизма приоритетного воспроизведения опыта.

В текущей работе данные о состоянии агента и динамических объектов объединяются в компактное и информативное представление, что позволяет сети принимать точные решения даже в условиях высокой плотности объектов на дороге. Такой подход особенно важен при обучении агента взаимодействию с большим количеством динамических участников движения, где необходимо учитывать множество факторов, включая расстояния, скорости и полосы движения объектов.

В первую очередь происходит обработка состояния агента, которое включает в себя пять параметров (x, y, скорость, номер полосы, габариты автомобиля в виде двух параметров). Оно обрабатывается линейным слоем с размерностью скрытого пространства 512. За линейным слоем следуют ReLU-активация и Dropout с вероятностью 10%, что помогает избежать переобучения.

Вместе с тем следует обработка состояния динамических объектов. Состояния каждого из 9 динамических объектов, включающие те же пять параметров, подаются на аналогичный линейный слой, что позволяет получить скрытое представление для каждого объекта.

Для учета значимости различных динамических объектов в текущей дорожной ситуации используется механизм внимания. На основе скрытых представлений объектов вычисляются весовые коэффициенты, определяющие вклад каждого объекта в финальное решение. Коэффициенты нормализуются с помощью Softmax, а затем используется операция взвешенного суммирования.

Полученные скрытые представления агента и агрегированные признаки динамических объектов конкатенируются и проходят через общий линейный слой с размерностью 512. Это позволяет объединить информацию о состоянии агента и о его окружении.

Финальное объединенное представление разделяется на две ветви:

- V(s) – функция ценности (Value Function) оценивает, насколько благоприятно текущее состояние независимо от конкретных действий;
- A(s,a) – функция преимущества (Advantage Function) определяет полезность каждого действия относительно среднего действия.

Эти два значения комбинируются для вычисления итоговой Q-функции, что позволяет алгоритму точнее выбирать действия в сложных ситуациях.

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s,a') \right), \quad (1)$$

где $|\mathcal{A}|$ — количество возможных действий.

B. Графовая нейронная сеть (GNN)

Графовые нейронные сети представляют мощный инструмент для моделирования взаимодействий между объектами в пространстве, что особенно полезно в задачах, требующих анализа сложных пространственных зависимостей. В данной работе используется GNN на основе Double DQN, которая представляет дорожную ситуацию в виде графа. Узлами графа являются агент и динамические объекты, а ребрами — расстояния между ними.

Архитектура сети состоит из трех графовых сверточных слоев (GCNConv), которые поочередно обрабатывают входные данные. На вход сети подается матрица признаков узлов графа, содержащая: координаты объекта (x, y), скорость (v), номер полосы ($lane$), размерность объекта ($size$). Каждый узел представляет состояние одного из участников движения, включая агента.

Кроме того, в сеть подается матрица индексов ребер, описывающая связи между объектами путем задания расстояния до ближайших объектов в пределах заданного радиуса видимости. Каждый графовый сверточный слой вычисляет новые признаки для узлов, учитывая информацию от соседних узлов через ребра графа.

Использование трех слоев позволяет последовательно обрабатывать взаимодействия между объектами, постепенно увеличивая охват соседних узлов. Соответственно, входные признаки узлов (x) и индексы ребер ($edge\_index$) проходят обработку через три уровня сверток (рисунок 1). Первый слой ($GCNConv$) извлекает базовые признаки объектов и их ближайших соседей. Второй слой учитывает взаимодействия между объектами на более высоком уровне. Третий слой формирует финальное представление каждого объекта, обобщая полученную информацию. На выходе получается обновленная матрица признаков узлов, которая используется для предсказания действий агента.

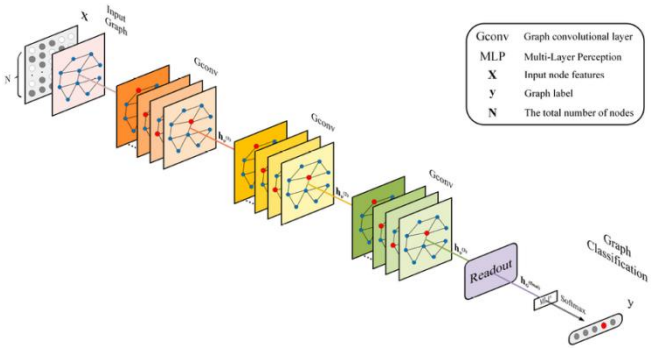


Рис. 3. Архитектура GNN

Основным преимуществом GNN является ее способность моделировать сложные зависимости между объектами на дороге, что особенно важно при высоком уровне плотности движения [14]. При этом GNN требует меньше вычислительных ресурсов на этапе вывода, что делает ее подходящей для применения в реальном времени [15]. Это связано с тем, что сеть обрабатывает лишь локальные взаимодействия между узлами, вместо анализа всего состояния целиком, как в методах с плотными тензорами.

IV. Сравнение

В данном разделе проведено подробное сравнение двух описанных ранее подходов: Rainbow DQN и графовой нейронной сети (GNN). Для обеих моделей было выполнено 100 эпизодов оценки. Сравнение охватывает различные аспекты их производительности, такие как эффективность обучения, динамика поведения агентов, управление наградами и штрафами, затраты вычислительных ресурсов, а также особенности финальных этапов обучения.

Таблица 1 отображает количественные оценки для двух подходов.

ТАБЛИЦА I. Оценка метрик

	RDQN	**GNN**
Средняя награда	771.98	1056.70
Минимальная награда	-4.29	-3.72
Максимальная награда	2506.46	2503.40
Средняя длина эпизода	198.51	243.27
Среднее время эпизода (сек.)	0.28	0.39
Количество аварий на 500 запусков	1.4 %	0.8 %

A. *Эффективность обучения*

Rainbow DQN и GNN показали схожую среднюю награду (2) в рамках эксперимента: 771.98 для Rainbow DQN и 1056.70 для GNN. Однако стоит отметить, что эти значения могут быть обусловлены разными стратегиями, выбранными каждой моделью для достижения целей. Rainbow DQN стремилась избегать штрафов за низкую скорость и излишнюю агрессивность, что сделало ее стратегию более универсальной и стабильной.

$$R_{avg} = \frac{1}{N} \sum_{i=1}^{N} R_i, \qquad (2)$$

где R_i — награда в i-м эпизоде, N — общее число эпизодов.

Максимальная награда Rainbow DQN достигла 2506.46, что демонстрирует ее способность находить оптимальные траектории и минимизировать штрафы. Минимальная награда составила -200, что обусловлено редкими эпизодами столкновений или критического снижения скорости.

Для GNN максимальная награда составила 2503.40, что близко к результату Rainbow DQN. Однако минимальная награда составила -513, что указывает на большую нестабильность модели в некоторых эпизодах.

Rainbow DQN быстрее достигала стабильных результатов при сравнении скорости обучения, тогда как GNN требовала большего количества итераций для выхода на сопоставимый уровень наград.

Для визуализации продленной работы сети были запущены в 2 средах с разным количеством машин: с 8 машинами и с 6 машинами.

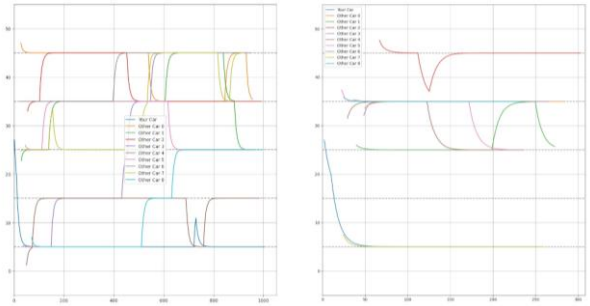


Рис. 4. Сравнение работы RDQN (слева) и GNN (справа) в среде с 8 машинами

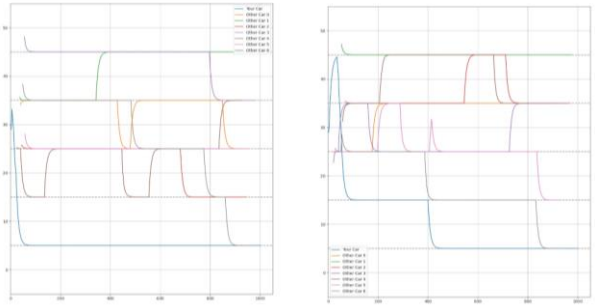


Рис. 5. Сравнение работы RDQN (слева) и GNN (справа) в среде с 6 машинами

Далее можно увидеть графики обучения моделей в подготовленной среде.

Рис. 6. График обучения RDQN

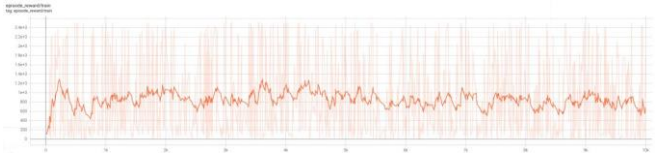


Рис. 7. График обучения GNN

B. Динамика поведения

Обе модели продемонстрировали способность адаптироваться к дорожной ситуации, снижая скорость перед препятствиями и удерживая ее на уровне потока, если присутствовали другие автомобили.

Rainbow DQN отличилась избеганием избыточного перестроения, предпочитая стабильное движение с минимальными рисками. При отсутствии других машин в окружающей среде сеть поддерживала среднюю скорость агента около 20 м/с, а в плотных условиях снижала скорость до 8-15 м/с, что свидетельствует о ее ориентировании на безопасность.

В свою очередь GNN в отдельных эпизодах проявляла нестабильность, резко снижая скорость после достижения нескольких сотен метров, что преждевременно заканчивало эпизод. Это могло быть связано с недостаточной проработкой механизмов долгосрочного планирования.

C. Награды и штрафы

Система наград задана следующим образом:

- +0.01 за каждый пройденный метр;
- +10 за каждые 1000 метров (окончание эпизода);
- -5 за столкновение (окончание эпизода);
- -20 за скорость менее 15 м/с;
- -200 за скорость менее 5 м/с (окончание эпизода).

Rainbow DQN эффективно балансировала между скоростью и безопасностью, избегая ситуаций, приводящих к штрафам за низкую скорость, а также стремилась минимизировать количество столкновений, что обеспечивало высокую стабильность.

GNN в отдельных эпизодах допускала резкое снижение скорости, что приводило к большим штрафам, и была более подвержена столкновениям, особенно в сложных сценариях с высокой плотностью трафика.

D. Затраты вычислительных ресурсов

Для Rainbow DQN среднее время обработки одного эпизода составило 0.28 секунд, что делает ее более быстрой в обработке, несмотря на сложность архитектуры. Более высокие вычислительные затраты оправданы за счет производительности и стабильности сети.

Среднее время обработки одного эпизода GNN составило 0.39 секунд. Несмотря на то, что архитектура GNN менее сложная, она требует больше времени на обработку данных. Меньшие затраты ресурсов делают ее потенциально более экономичной, но это сопровождается меньшей стабильностью результатов.

E. Результаты финальных чекпоинтов

На поздних этапах обучения обе модели демонстрировали высокие результаты, однако Rainbow DQN отличалась большей стабильностью.

Последние чекпоинты Rainbow DQN достигли наград порядка 3500 ±90. Это свидетельствует о высокой надежности и способности модели адаптироваться к различным сценариям.

Для GNN последние чекпоинты также достигли наград порядка 3500, но с большим разбросом результатов, что указывает на нестабильность в отдельных эпизодах.

V. Заключение

Каждая сеть рассмотрена с точки зрения ее архитектуры, процесса обучения и особенностей реализации в рамках поставленной задачи. Приведенные подходы были сравнены по производительности, стабильности и поведению в симулированной среде.

Rainbow DQN, в целом, демонстрирует более высокую производительность и стабильность, что делает ее предпочтительным выбором для задач, требующих точного планирования и адаптивности. Тем не менее, результаты GNN показывают перспективы ее дальнейшего улучшения, например, за счет более глубокого анализа гиперпараметров или увеличения времени обучения. По этой причине GNN может оказаться полезной в условиях ограниченных вычислительных ресурсов, где ее сравнительно низкая стоимость обучения может компенсировать некоторую потерю в стабильности поведения. Дополнительно оптимизация архитектуры модели и исследование новых подходов к работе с графовыми данными могут позволить GNN выйти на уровень, сравнимый с Rainbow DQN.

Литература

[1] Mnih, V., Kavukcuoglu, K., Silver, D., et al. "Playing Atari with Deep Reinforcement Learning". arXiv preprint arXiv:1312.5602, 2013.

[2] Hessel, M., Modayil, J., Van Hasselt, H., et al. "Rainbow: Combining Improvements in Deep Reinforcement Learning". arXiv preprint arXiv:1710.02298, 2017.

[3] Абакумов, А. А. Вопросы сегментации дорожного слоя / А. А. Абакумов, В. О. Хуако // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 40-45. – EDN UWTAKJ.

[4] Грищенко, Д. И. Классификация земного покрова и землепользования / Д. И. Грищенко // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 30-35. – EDN AYJWGA.

[5] Формирование траектории корреляционно-экстремальной навигационной системы по критерию минимума погрешностей координат / А. В. Шолохов, С. Б. Беркович, Н. И. Котов, Р. Н. Садеков // Юбилейная XXV Санкт-Петербургская Международная конференция по интегрированным навигационным системам : Сборник материалов, Санкт-Петербург, 28–30 мая 2018 года / Главный редактор В.Г. Пешехонов. – Санкт-Петербург: "Концерн "Центральный научно-исследовательский институт "Электроприбор", 2018. – С. 175-177. – EDN UZJVXO.

[6] Использование 3D-сетей для «предсказания» моделей поведения транспортных средств в задаче беспилотного движения трамвая / Н. С. Гужва, В. Е. Прун, В. В. Постников [и др.] // XXIX Санкт-Петербургская международная конференция по интегрированным навигационным системам : сборник материалов, Санкт-Петербург, 30 мая – 01 2022 года. – Санкт-Петербург: "Концерн "Центральный научно-исследовательский институт "Электроприбор", 2022. – С. 304-310. – EDN JQNIIU.

[7] Kipf, T. N., Welling, M. "Semi-Supervised Classification with Graph Convolutional Networks". arXiv preprint arXiv:1609.02907, 2016.

[8] Li, Y., et al. "Graph Neural Networks for Reinforcement Learning". arXiv preprint arXiv:1806.01261, 2018.

[9] Kingma, D. P., Ba, J. "Adam: A Method for Stochastic Optimization". arXiv preprint arXiv:1412.6980, 2014.

[10] Peng, B., Yan, X., Duchi, J., et al. "Learning Markov Games with Independent Transition Functions." arXiv preprint arXiv:2011.14826, 2020.

[11] Zhou, C., Huang, L., Liao, H., et al. "Distributional Soft Actor-Critic for Risk-Sensitive Reinforcement Learning." arXiv preprint arXiv:2004.13291, 2020.

[12] Lillicrap, T. P., et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).

[13] Sutton, R. S., & Barto, A. G. "Reinforcement Learning: An Introduction." MIT Press, 2nd Edition, 2018.

[14] Mnih, V., Kavukcuoglu, K., Silver, D., et al. "Human-level control through deep reinforcement learning." Nature, 518, 529–533 (2015). DOI: 10.1038/nature14236.

[15] Liu, M., Han, K., and Shi, L. "Learning Interactive Agents for Multi-Agent Tracking with Graph Neural Networks." arXiv preprint arXiv:2009.11905, 2020.

ГРНТИ 28.23.37
ASJC 1702
OECD 01.02.00

Обнаружение и классификация повреждений костей с использованием нейронных сетей

И. А. Коротких
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2008358@edu.misis.ru

***Аннотация*— в настоящее время широкое распространение получили нейронные сети, специализирующиеся на определении и распознавании объектов. Все чаще поднимаются вопросы возможности практического применения подобных нейросетевых алгоритмов в узкоспециализированных отраслях для решения специфических задач вместо или вместе с человеком. Одной из возможных сфер применения подобного класса нейросетей является медицина. В работе рассматривается возможность переобучения нейросетевых алгоритмов YOLOv11 и DETR для решения задачи определения повреждения костей по рентгеновским снимкам. Такое обучение предполагает решение задач, связанных с распознаванием различных видов травм: переломов, вывихов. трещин и других повреждений костей. Обученная нейросеть должна быть способна воспринимать все анатомические особенности человека, чтобы иметь возможность выносить качественные диагнозы. В работе представлены результаты обучения нейросетей на наборе данных Roboflow.**

Ключевые слова — компьютерное зрение, классификация изображений, сегментация изображений, классификация объектов, детекция переломов, распознавание переломов, нейродоктор, YOLO, DETR, Roboflow.

I. Введение

В последние десятилетия происходит активное развитие и внедрение технологий искусственного интеллекта. Он способен оказать значительное влияние на многие сферы жизни людей, ИИ изменяет производственные процессы, экономическую структуру, затрагивает повседневные взаимодействия, здравоохранение, образование и многие другие сферы. Одним из главных направлений применения ИИ на данный момент считают развитие дронов и БПЛА [1] и транспортной сферы [2].

Искусственный интеллект широко применяется и имеет значительный потенциал дальнейшего внедрения в сфере медицины. ИИ имеет возможность качественно и быстро предсказывать диагноз по изображениям организма пациента. Снимок организма является источником важнейшей информации о состоянии здоровья человека, поэтому необходимо максимально точно читать изображения и принимать во внимание все возможные особенности пациента. Даже для опытного специалиста точный анализ снимка является тяжелой задачей. Специально обученная на базе данных нейросеть способна в автоматическом режиме обрабатывать десятки снимков, с принятием во внимание всех особенностей пациента, значительно разгружая работу докторов [3].

Нейросетевые алгоритмы могут использоваться для решения широкого спектра медицинских задач, от сегментации медицинских изображений [4] до более сложных задач, таких как классификация катаракты глаза [5]. Одним из наиболее перспективных для внедрения нейросетевых алгоритмов разделов медицины является травматология. Определение переломов, вывихов, сдвигов и трещин является вполне реализуемой задачей для современных нейросетей.

При создании системы определения повреждения костей важными задачами являются обнаружение и распознавание различных травм. Для решения применяются технологии компьютерного зрения. В числе задач по обнаружению/распознаванию объектов на изображениях, полученных при рентгене поврежденного участка скелета, можно выделить задачу идентификации и классификации повреждений кости.

Обнаружение и распознавание той или иной травмы включает в себя непосредственно её обнаружение и классификацию (перелом, вывихи или трещина). В литературе описаны различные способы решения этой задачи как с точки зрения машинного обучения [6], так и с медицинской стороны вопроса [7].

Методы глубокого обучения показали высокую производительность и способность к обобщению во многих областях и типах задач, таких как классификация [8] и обнаружение [9]. Детекторы объектов общего назначения были хорошо изучены для задач, связанных с обнаружением и классификацией различных объектов как на отдельных изображениях, так и на целых видео. YOLO [10] и DETR [11] являются современными детекторами. В работе рассматриваются и сравниваются возможности нейросетей по определению и классификации изображений. Сравниваются результаты обучения и тестирования данных нейросетевых алгоритмов для решения задачи определения повреждения костей.

Подходы, основанные на обучении, особенно те, которые используют глубокое обучение, требуют больших объёмов аннотированных данных, что не всегда есть в наличии. В настоящее время в свободном доступе есть зарубежные базы изображений с

аннотациями повреждений костей, для обучения нейросетей в этой работе будет использоваться база данных ресурса Roboflow.

II. НАБОРЫ ДАННЫХ

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались открытые ~~и~~ заранее размеченные наборы данных. Рассмотрим используемый открытый набор.

Roboflow 100: bone fracture computer vision project

Данный набор общедоступных данных, основанный на рентгеновских изображениях реальных травмированных и здоровых частей тела, собранных Jason Zhang и Caden Li для собственного исследовательского проекта по машинному обучению, содержит несколько сотен аннотированных изображений рентгеновских снимков различных частей тела (рисунок 1-2). В наборе данных представлены различные варианты травм – всего 4 класса (рисунок 3): *angle*, *fracture*, *line*, *messed_up_angle.*

Также в наборе данных представлены некоторые нестандартные вариации изображений с лишними визуальными элементами и случайным наклоном (рисунок 4).

Рис. 1. Пример перелома кости ноги

Рис. 2. Пример смещения пальцев руки

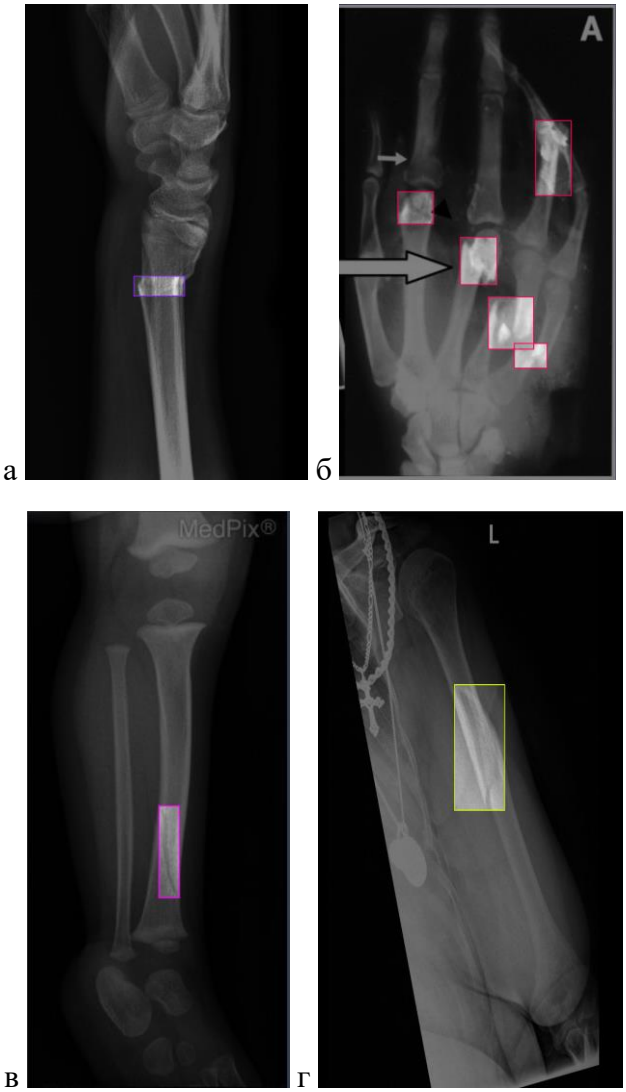


Рис. 3. Примеры классов повреждения костей: а) angle, б) fracture, в) line, г) messed_up_angle

Рис. 4. Пример нестандартного изображения

III. НЕЙРОСЕТЕВЫЕ АЛГОРИТМЫ

Целью данной работы является решение задачи обнаружения и классификации различных видов повреждения кости для интеграции в сферу медицины. Данный тип задачи уже решался ранее [12-13], в работах авторы предлагают разработку новой системы обнаружения повреждений костей, основываясь на алгоритмах SIFT и трансформерах Haar Wavelet [12] и использование отдельных алгоритмов машинного обучения [13]. Данные работы предлагают разработку новых алгоритмов обучения (рисунок 5-6) и последующее создание новых нейросетей, заточенных конкретно для решения данной задачи.

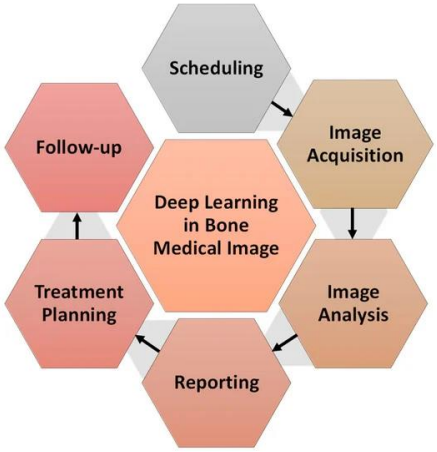


Рис. 5. Процесс машинного обучения в радиологии

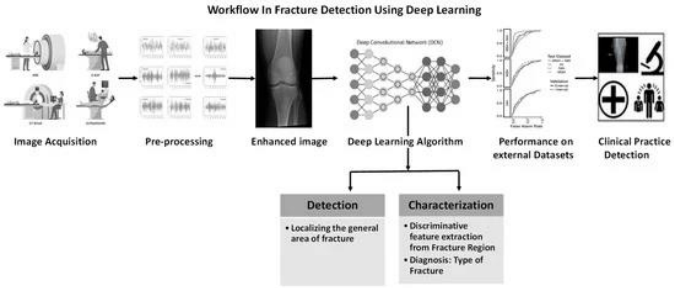


Рис. 6. Алгоритм обнаружения повреждений кости, используя методику машинного обучения

Данный подход к решению поставленной задачи, в рамках данной работы, видится избыточным и не оптимальным, так как он предполагает разработку и обучение совершенно нового алгоритма с нуля. Разработка и успешное применение нового алгоритма для определения объектов на изображении является достаточно емкой задачей, не учитывая последующее обучение для решения задачи определения повреждений кости. Уже сейчас существует широкое разнообразие моделей детекции и классификации изображений с открытым кодом, которые можно обучить на необходимом датасете и внедрить в нуждающуюся предметную область для решения поставленной задачи. При использовании подобных моделей можно пропустить стадию разработки, которая является избыточной для поставленной задачи в рамках данной статьи, и сконцентрировать усилия на анализе возможности решения задачи детекции повреждений кости.

1. *YOLOv11*

YOLOv11 (You only look once) [14] – версия серии нейросетевых моделей детекции изображений. YOLOv11 отличается своей повышенной адаптивностью, поддерживая расширенный спектр задач компьютерного зрения (CV), выходящих за рамки традиционного обнаружения объектов. Среди них выделяются оценка позы и сегментация экземпляров, что расширяет применимость модели в различных областях.

В своей основе архитектура YOLO состоит из трех фундаментальных компонентов. Во-первых, основа (backbone) служит основным экстрактивным элементом, используя сверточные нейронные сети для преобразования необработанных данных изображения в многомасштабные карты признаков. Во-вторых, компонент шеи (neck) выполняет роль промежуточной стадии обработки, используя специализированные слои для агрегации и улучшения представлений признаков на разных масштабах. В-третьих, компонент головы (head) функционирует как механизм предсказания, генерируя конечные выходные данные для локализации и классификации объектов на основе уточненных карт признаков. Основываясь на этой устоявшейся архитектуре, YOLOv11 расширяет и улучшает основы, заложенные в YOLOv8, вводя архитектурные новшества и оптимизации параметров для достижения превосходной производительности обнаружения, как показано на рисунке 7.

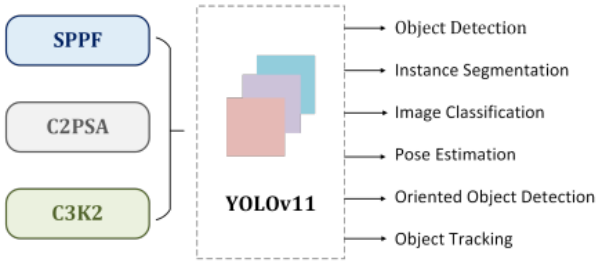


Рис. 7. Ключевые архитектурные модули YOLOv11

Также в процесс обучения были включены: аугментация изображений при помощи изменения оттенка, насыщенности, экспозиции, батч-нормализация. Кроме того, каждые 10 батчей менялось разрешение изображений с 608x608 на разрешения, кратные 32, что делает модель более устойчивой к разным масштабам.

Backbone является ключевым компонентом, ответственным за извлечение признаков из входного изображения. Этот процесс включает в себя наслоение сверточных слоев и специализированных блоков для генерации карт признаков на различных разрешениях. YOLOv11 использует блок C3k2 [15] для обработки информации. Блок C3k2 представляет собой более вычислительно эффективную реализацию частичного узкого места промежуточной стадии (Cross Stage Partial, CSP). Он использует две меньшие свертки вместо одной

крупной. YOLOv11 сохраняет блок пространственной пирамидальной агрегации (Spatial Pyramid Pooling - Fast, SPPF) из предыдущих версий, но вводит новый блок Cross Stage Partial с пространственным вниманием (C2PSA) после него [15]. Путем пространственной агрегации признаков блок C2PSA позволяет YOLOv11 сосредоточиться на конкретных областях интереса, что улучшает точность детекции для объектов различных размеров и положений.

Шея (neck) объединяет признаки на разных масштабах и передает их в голову (head) для предсказания. YOLOv11 использует блок C3k2 в шее.

Голова (head) YOLOv11 отвечает за генерацию окончательных предсказаний в терминах обнаружения и классификации объектов. Она обрабатывает карты признаков, переданные из шеи, выводя ограничивающие рамки и метки классов для объектов на изображении. В секции головы YOLOv11 использует несколько блоков C3k2 для эффективной обработки и уточнения карт признаков. Голова YOLOv11 включает несколько слоев CBS (Convolution-BatchNorm-Silu) [16] после блоков C3k2.

Эти слои дополнительно уточняют карты признаков, выполняя следующие задачи:

- Извлечение релевантных признаков для точного обнаружения объектов.
- Стабилизация и нормализация потока данных с помощью пакетной нормализации.
- Использование функции активации Sigmoid Linear Unit (SiLU) для введения нелинейности, что улучшает производительность модели.

Блоки CBS служат основными компонентами как в извлечении признаков, так и в процессе детекции, обеспечивая передачу уточненных карт признаков на последующие слои для предсказания ограничивающих рамок и классификации. Каждая ветвь детекции заканчивается набором слоев Conv2D, которые уменьшают количество признаков до необходимого числа выходов для координат ограничивающей рамки и предсказаний классов. Финальный слой Detect объединяет эти предсказания, которые включают:

- Координаты ограничивающих рамок для локализации объектов на изображении.
- Оценки наличия объектов (objectness scores), указывающие на наличие объектов.
- Оценки классов для определения класса обнаруженного объекта.

2. DETR - Detection Transformer

DETR – End-to-end Object Detection with Transformers является новой структурой обработки изображений для обнаружения объектов. Основными компонентами DETR являются основанная на множестве глобальная функция потерь, которая обеспечивает уникальные предсказания через двусторонние соответствия, и архитектура кодировщика-декодировщика библиотеки transformers [17]. Имея фиксированный небольшой набор изученных объектов и запросов, DETR анализирует взаимосвязи объектов и глобальный контекст изображения, чтобы напрямую выводить финальный набор предсказаний параллельно. Новая модель концептуально проста и не требует специализированной библиотеки, в отличие от многих других современных детекторов. DETR демонстрирует высокую точность и скорость работы на уровне хорошо зарекомендовавшей себя и высоко оптимизированной базовой модели Faster R-CNN на сложном наборе данных COCO. DETR легко обобщается для продуктивного сегментирования в унифицированном формате [11].

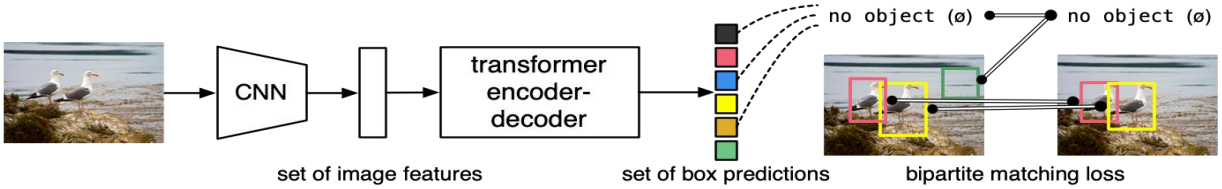


Рис. 8. Архитектура DETR

DETR выводит фиксированный набор из N прогнозов за один проход через декодер, где N устанавливается значительно больше, чем количество объектов на изображении. Одной из основных трудностей обучения является оценка предсказанных объектов (класс, позиция, размер) относительно истинных значений. Функция потерь DETR создает оптимальное двустороннее соответствие между предсказанными и истинными объектами, а затем оптимизирует специфические для объектов (ограничивающие рамки) потери.

Архитектура DETR достаточно проста (рисунок 8). Она состоит из трех основных компонентов: основной CNN, трансформер с кодером и декодером, а также простая полносвязанная сеть (FFN), которая делает окончательное предсказание обнаружения.

Начав с исходного изображения с 3 цветными каналами, стандартная CNN-основа генерирует карту активации с более низким разрешением. Типичные используемые значения составляют C=2048 и H,W = $\frac{H0}{32}, \frac{W0}{32}$

Сначала свертка 1x1 уменьшает размерность канала высокой активации карты f с C до меньшей размерности d, создавая новую карту признаков z0. Кодер ожидает последовательность на вход, поэтому мы объединяем пространственные размеры z0 в одно измерение, в результате чего получается карта признаков d*HW. Каждый слой кодера имеет стандартную архитектуру и состоит из многоголового модуля самовнимания (self-attention) и полносвязанной сети (FFN).

Декодер следует стандартной архитектуре трансформера, преобразуя N встраиваний размера d, используя механизмы многоголового самовнимания и внимания кодера-декодера. N объектных запросов преобразуются в выходное встраивание декодером.

Затем они независимо декодируются в координаты рамок и метки классов с помощью полносвязанной сети, в результате чего получается N окончательных предсказаний.

Окончательное предсказание вычисляется с помощью перцептрона с 3 слоями с функцией активации ReLU и скрытой размерностью d и линейным проекционным слоем.

IV. Сравнение

Для сравнения нейросетевых алгоритмов YOLOv11 и DETR в способности решения задачи определения и классификации повреждения костей было проведено их развертывание в Google Collaboratory. Обе нейросети были обучены на одном и том же наборе данных, представленном 326 тренировочными, 88 валидационными и 44 тестовыми изображениями рентгеновских снимков различных частей тела, как с повреждениями, так и без. Набор данных имеет 4 класса: angle, fracture, line и messed_up_angle. Обе нейросети прошли обучение в 200 epoch. Результаты обучения для YOLOv11 представлены на рисунках 9-12.

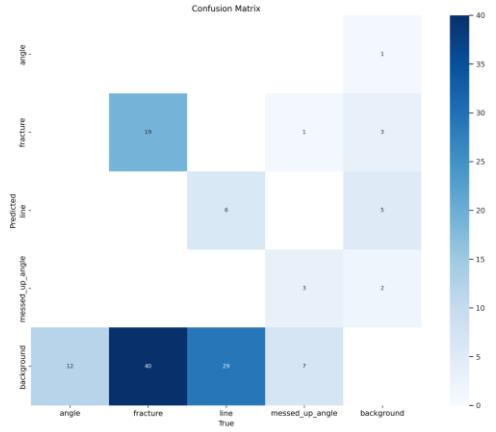


Рис. 9. Confusion Matrix

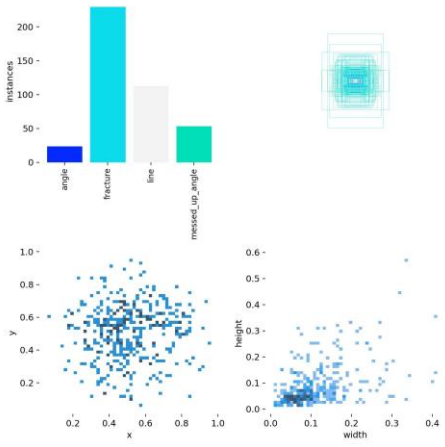


Рис.10. Labels

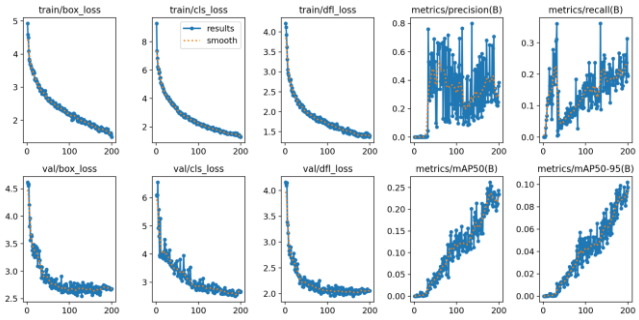


Рис. 11. Графики результатов обучения

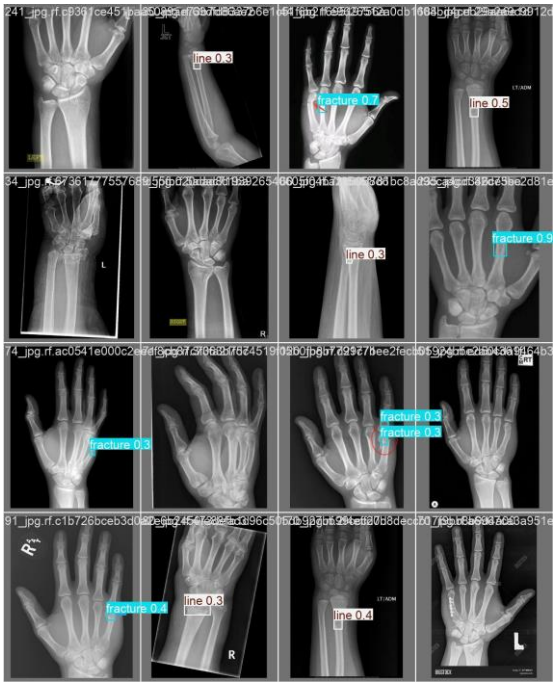


Рис. 12. Пример предсказаний

Алгоритм внедрения нейросети DETR не включает в себя вывод такого же подробного набора данных и дает только результаты обнаружения (рисунок 13-15).

Рис.13. Обнаружение fracture

Рис. 14. Обнаружение messed_up_angle

Рис. 15. Обнаружение angle

Полученные результаты представлены в таблице 1.

Таблица 1. Результаты обучения

	YOLOv11			DETR		
	box loss	cls loss	dfl loss	box loss	cls loss	dfl loss
epoch 0-20	4.9-3.2	9.3-4.4	4.2-2.4	5.1-3.5	8.9-5.0	4.6-3.5
epoch 21-40	3.2-2.8	4.4-3.6	2.4-2.2	3.4-2.9	5.0-4.3	3.4-3.1
epoch 41-60	2.8-2.6	3.5-3.0	2.1-1.9	2.9-2.6	4.2-4.0	3.0-2.7
epoch 61-80	2.5-2.4	2.9-2.5	1.9-1.8	2.6-2.5	3.9-3.6	2.6-2.4
epoch 81-100	2.3-2.2	2.5-2.3	1.8-1.7	2.4-2.3	3.6-3.3	2.4-2.1
epoch 101-120	2.1-2.1	2.3-2.1	1.6-1.6	2.2-2.1	3.2-2.9	2.0-1.8
epoch 121-140	2.0-1.9	1.9-1.7	1.6-1.5	2.0-1.9	2.8-2.5	1.8-1.7
epoch 141-160	1.9-1.8	1.8-1.6	1.5-1.5	1.8-1.8	2.4-2.1	1.7-1.7
epoch 161-180	1.8-1.7	1.6-1.5	1.4-1.4	1.7-1.6	2.1-1.9	1.7-1.6
epoch 181-200	1.7-1.5	1.5-1.3	1.4-1.4	1.6-1.5	1.8-1.7	1.6-1.6
Среднее время на одну epoch	~11 секунд			~3,4 минуты		

Сравнив результаты обучения двух алгоритмов, можно сделать два вывода. Для обеих моделей нейросетевых алгоритмов тренировки в 200 epoch не являются достаточными, для того чтобы показывать достаточную уверенность в своих прогнозах. Обе нейросети дают как достаточно высокие показатели уверенности предсказаний (от 0.8 до 0.96), так и низкие. Показатели loss для обеих нейросетей примерно равны, но YOLOv11 показывает чуть лучшие показатели по ходу и в итоге обучения. Также стоит отметить, что процесс её обучения занял кратно меньшее время: в среднем 1 epoch у YOLOv11 занимала 11 секунд, а DETR 3 минуты. Дополнительно, сам процесс развертки YOLOv11 является значительно более простым и быстрым. Для ее работы необходима только установка библиотеки ultralitycs, в то время как DETR требует сразу несколько предустановленных библиотек, которые занимают достаточно много места на устройстве (~15-20 гб) и отдельный код.

Провести полное обучение для получения предсказаний со стабильно высоким уровнем уверенности не является возможным из-за ограниченных ресурсов Google Collaboratory. Основываясь на наблюдениях во время тренировки нейросетей, можно обратить внимание на качественное улучшение предсказаний по мере увеличения количества тренировочных epoch. Исходя из этого, можно сделать вывод, что обе нейросети, при достаточном уровне обучения, пригодны для решения задачи обнаружения повреждения костей. При равных показателях обучения, вероятно, стоит сделать выбор в пользу YOLOv11 в силу большей простоты

развертывания и значительно более высокой скорости обучения.

V. Заключение

Были рассмотрены основные наборы данных, на которых обучались и тестировались рассматриваемые нейронные сети. Приведены два подхода к детектированию и классификации видов повреждений костей: YOLOv11 и DETR. Каждая сеть рассмотрена с точки зрения её архитектуры, процесса обучения, используемых для обучения и тестирования наборов данных.

Приведённые подходы были сравнены на открытом датасете поврежденных костей из Roboflow. В конце сделан вывод, что обе нейросети пригодны для решения поставленной задачи, но YOLOv11 имеет преимущества в виде более простого процесса развертывания и более быстрого процесса обучения.

Литература

[1] Али Б., Садеков Р. Н., Цодокова В. В. Алгоритмы навигации беспилотных летательных аппаратов с использованием систем технического зрения //Гироскопия и навигация. – 2022. – Т. 30. – №. 4 (119). – С. 87.

[2] Жебрак Л. М. и др. РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА АЭРОФОТОСНИМКАХ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧАХ МАРШРУТНОЙ НАВИГАЦИИ. – 2021.

[3] Елизарова М. И. и др. Искусственный интеллект в медицине //International Journal of Professional Science. – 2021. – №. 5. – С. 81-85

[4] Исаченко, М. К. Сегментация медицинских изображений с помощью DUCK-Net / М. К. Исаченко, Р. Б. Парчиев // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 54-60. – EDN VWUJOG.

[5] Мельникова, М. Ф. Классификация катаракты глаза при помощи компьютерного зрения / М. Ф. Мельникова // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 100-104. – EDN XEDDEM.

[6] Tanzi L. et al. X-ray bone fracture classification using deep learning: a baseline for designing a reliable approach //Applied Sciences. – 2020. – Т. 10. – №. 4. – С. 1507.

[7] Ishman S. L., Friedland D. R. Temporal bone fractures: traditional classification and clinical relevance //The Laryngoscope. – 2004. – Т. 114. – №. 10. – С. 1734-1741.

[8] R. F. Berriel, A. T. Lopes, A. F. de Souza, and T. Oliveira-Santos, “Deep Learning Based Large-Scale Automatic Satellite Crosswalk Classification,” IEEE Geoscience and Remote Sensing Letters, vol. 14, pp. 1513–1517, Sept 2017.

[9] R. Guidolini, L. G. Scart, L. F. R. Jesus, V. B. Cardoso, C. Badue, and T. Oliveira-Santos, “Handling Pedestrians in Crosswalks Using Deep Neural Networks in the IARA Autonomous Car,” in 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, July 2018.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

[11] Carion N. et al. End-to-end object detection with transformers //European conference on computer vision. – Cham : Springer International Publishing, 2020. – С. 213-229.

[12] Dimililer K. IBFDS: Intelligent bone fracture detection system //Procedia computer science. – 2017. – Т. 120. – С. 260-267.

[13] Meena T., Roy S. Bone fracture detection using deep supervised learning from radiological images: A paradigm shift //Diagnostics. – 2022. – Т. 12. – №. 10. – С. 2420.

[14] Alif M. A. R. YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems //arXiv preprint arXiv:2410.22898. – 2024.

[15] Satya Mallick. Yolo - learnopencv. https://learnopencv.com/yolo11/, 2024. Дата обращения: 25.12.2024.

[16] Jingwen Feng, Qiaofeng An, Jiahao Zhang, Shuxun Zhou, Guangwei Du, and Kai Yang. Application of yolov7-tiny in the detection of steel surface defects. In 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), pages 2241–2245. IEEE, 2024.

[17] Dosovitskiy A. An image is worth 16x16 words: Transformers for image recognition at scale //arXiv preprint arXiv:2010.11929. – 2020.

ГРНТИ 28.23.37
ASJC 1707
OECD 1.02

Детектирование диких животных при помощи нейронных сетей

Ю.А. Криворот
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m1805775@edu.misis.ru

Е.А. Ильяков
кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2412418@edu.misis.ru

В данной работе исследуется эффективность нейронных сетей YOLOv8 и Faster R-CNN для решения задачи детекции животных на изображениях и видео. Модели обучались на уникальном датасете, сформированном и размеченном автором, с использованием данных из публичных источников, таких как платформа Kaggle. Для оценки производительности моделей были применены метрики, характерные для задач детекции и классификации объектов. На основе полученных результатов был сделан вывод о сильных сторонах обеих моделей, а также о сравнении их производительности в зависимости от специфики задачи.

Ключевые слова — компьютерное зрение, детекция объектов, классификация, детекция животных, YOLO, Faster R-CNN, глубокое обучение.

I. Введение

В современном мире, где технологии искусственного интеллекта стремительно развиваются, автоматическое распознавание объектов на изображениях и видео играет все более важную роль в различных областях, от мониторинга биоразнообразия и охраны дикой природы до автоматизации зоологических исследований [1]. Одним из ключевых направлений в этой области является детекция и классификация животных на изображениях и видео. Автоматизированные системы, способные точно идентифицировать животных, открывают широкие перспективы для решения ряда практических задач, включая учет популяций, контроль за миграцией животных, а также предотвращение браконьерства.

Современные методы глубокого обучения продемонстрировали впечатляющие результаты в задачах компьютерного зрения, включая классификацию и детекцию объектов [2, 3, 4]. Благодаря способности извлекать сложные признаки из изображений, нейронные сети обладают высоким потенциалом для точной идентификации животных в разнообразных условиях.

В контексте развития технологий глубокого обучения [5] выбор оптимальной архитектуры нейронной сети для задачи детекции и классификации животных является критически важным [6]. Среди множества существующих архитектур YOLO представляет собой один из самых популярных подходов к решению данной задачи [7, 8]. Архитектура YOLO отличается высокой скоростью работы, что делает ее привлекательной для приложений, требующих обработки большого объема данных в реальном времени. Точность её работы также не уступает другим моделям [9].

Данное исследование посвящено анализу эффективности нейронной сети YOLO и сравнение с нейронной сетью Faster R-CNN для задачи детекции и классификации диких животных.

II. Наборы данных

С целью проведения процессов обучения и тестирования рассматриваемой в данном исследовании нейронной сети автором был собран и собственноручно размечен набор данных.

Рассмотрим данные, использованные для составления набора, более детально.

A. Данные с платформы Kaggle

Обширный набор данных [10], представленный на платформе Kaggle, содержит 5400 изображений, охватывающих 90 классов различных животных. Эти изображения были собраны с ресурса Google Images. Структура датасета организована в виде архива, состоящего из отдельных папок, каждая из которых содержит изображения, принадлежащие к определенному классу. Датасет характеризуется сбалансированностью классов: каждый класс представлен равным количеством изображений. Для целей данного исследования из набора была отобрана подвыборка, включающая 10 классов и 600 изображений (по 60 изображений на каждый класс). Эта подвыборка была использована для обучения рассматриваемых моделей. Изначально данный датасет не содержал разметки для задач обнаружения объектов. В связи с этим авторами была выполнена ручная разметка выбранных 600 изображений с использованием специализированного инструмента для аннотирования изображений CVAT.

Примеры размеченных изображений представлены на рисунке 1.

а б

в г

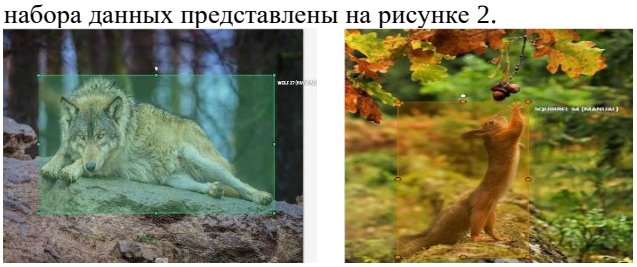
Рис. 1. Экземпляры классов из набора данных: а) барсук б) северный олень, в) олень, г) лиса

B. Собственный набор данных

В дополнение к данным с платформы Kaggle в работе также используется локальный набор данных, собранный автором исследования. Этот набор содержит фотографии животных, принадлежащих к тем же десяти классам, что и выборка из Kaggle. Учитывая, что 600 изображений с Kaggle недостаточно для полноценной валидации и тестирования моделей, автор дополнительно собрал и разметил по 100 изображений для каждой из этих задач. Изображения для локального набора данных были получены с ресурса Яндекс.Картинки.

Таким образом, для валидации и тестирования было использовано по 100 изображений соответственно. Разметка этих изображений также была выполнена вручную с использованием инструмента CVAT.

Примеры размеченных изображений из локального набора данных представлены на рисунке 2.



а б

в г

Рис. 2. Экземпляры классов из локально собранного набора данных: а) волк б) белка в) заяц г) ёж

III. Нейросетевые архитектуры

A. YOLOv8

YOLOv8, являясь последней итерацией семейства детекторов объектов YOLO (You Only Look Once), представляет собой мощную и эффективную архитектуру, построенную на трех ключевых компонентах: Backbone, Neck и Head. В отличие от своих предшественников, YOLOv8 предлагает ряд архитектурных улучшений, обеспечивающих более высокую скорость и точность обнаружения.

- Backbone (Основа) . Задача Backbone — извлекать иерархические признаки из входного изображения. Процесс начинается с захвата простых паттернов, таких как края и текстуры, на начальных уровнях сети. По мере углубления в сеть формируются все более абстрактные и сложные признаки, представляющие собой высокоуровневую семантическую информацию об изображении. YOLOv8 использует модифицированную и более эффективную версию CSPDarknet53 [11], где CSP (cross-stage partial connections) играют ключевую роль. CSP-соединения разделяют входной feature map на две части, одна из которых проходит через блок свертки, а другая — нет. Затем эти части объединяются, что способствует более эффективному градиентному потоку и уменьшению вычислительной сложности. Это позволяет сети обучаться быстрее и достигать лучшей производительности.

- Neck (Шея). Neck — это агрегатор признаков, связывающий Backbone и Head. Его основная функция — комбинировать признаки разных масштабов, извлеченные Backbone' для эффективного обнаружения объектов различных размеров. Вместо Feature Pyramid Network (FPN), которая широко используется в других архитектурах обнаружения объектов, YOLOv8 применяет новый модуль C2f (Concat and Convolutions from Feature maps) [12]. C2f более эффективно объединяет высокоуровневые семантические признаки с низкоуровневыми пространственными признаками, что значительно улучшает качество обнаружения, особенно для маленьких объектов. Этот процесс агрегации признаков играет решающую роль в способности YOLOv8 обнаруживать объекты на разных уровнях детализации и в различных контекстах. Интеграция контекстной информации в Neck также способствует повышению точности обнаружения.

- Head (Голова). Head отвечает за финальную стадию обнаружения объектов. Он принимает агрегированные признаки от Neck и генерирует предсказания: координаты ограничивающих рамок (bounding boxes), оценки достоверности (confidence scores) и классификацию обнаруженных объектов. YOLOv8 использует набор модулей детекции, работающих параллельно на выходе Neck. Предсказания этих модулей затем агрегируются для получения окончательного результата. Эта многоуровневая структура Head обеспечивает высокую точность и эффективность обнаружения объектов различных классов. Более того, в YOLOv8 используется decoupled head, разделяющий классификацию и регрессию, что способствует улучшению производительности [13, 14].

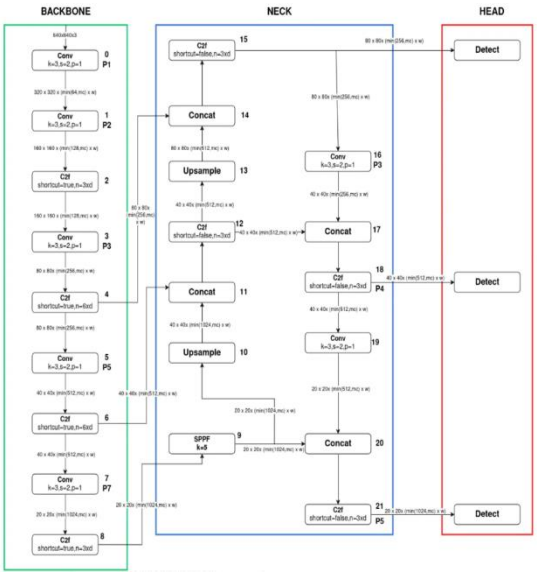


Рис. 3 . Архитектура YOLOv8

B. *Faster R-CNN*

Faster R-CNN (полное название — Faster Region-based Convolutional Neural Network) — это одна из наиболее эффективных и продвинутых моделей для обнаружения объектов. Она стала логичным продолжением моделей R-CNN и Fast R-CNN, внеся ключевое улучшение: интеграцию Region Proposal Network (RPN) для эффективного генерирования предложений областей (region proposals).

- Backbone
- Region Proposal Network (RPN). RPN отвечает за генерацию областей, которые, вероятно, содержат объекты. Вместо использования внешних алгоритмов (как в R-CNN), эта сеть встроена в архитектуру Faster R-CNN, что делает процесс более быстрым и точным. Сперва RPN создаёт набор фиксированных рамок (анкоров) различных масштабов и соотношений сторон. Эти рамки накладываются на каждую позицию сетки признаков, сформированной сверточной базой. Далее каждой анкорной рамке назначается вероятность наличия объекта и метка "объект" или "фон", после чего предсказанные рамки уточняются путём корректировки их координат, чтобы лучше охватить объекты. После фильтрации (например, с использованием non-maximum suppression) RPN передаёт несколько сотен лучших предложений (region proposals) для дальнейшей обработки.
- Классификатор и регрессор рамок. Этот модуль уточняет и классифицирует области, предложенные RPN. Основная задача — определить, к какому классу относится объект в рамке, и уточнить её координаты. Первым этапом идет преобразование в области фиксированного размера для обработки на последующих уровнях сети. ROI Align (более современный метод) обеспечивает большую точность, устраняя эффекты квантования, присущие ROI Pooling. Далее сеть определяет класс объекта. Параллельно выполняется уточнение координат рамки, чтобы лучше охватить объект.

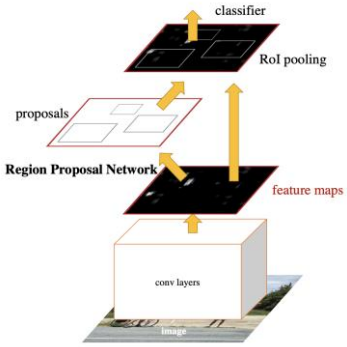


Рис. 4 . Архитектура Faster RCNN

IV. Проведенные исследования и результаты

A. *Обучение модели YOLOv8*

Обучение модели YOLOv8 для задачи обнаружения объектов на изображениях представляет собой ключевой этап в разработке эффективного решения для классификации и детекции.

Для обучения модели YOLOv8 была сформирована обучающая выборка, состоящая из 600 изображений, охватывающих 10 различных классов животных. Каждый экземпляр в наборе данных был аннотирован.

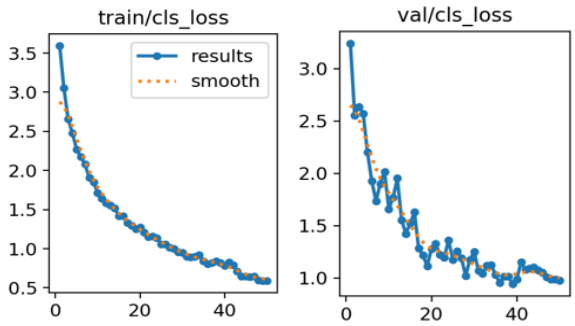
Для оценки использовалась валидационная выборка, состоящая из 100 изображений, которые не использовались в процессе обучения. Обучение модели проводилось на протяжении нескольких этапов, включающих в себя 50, 100, 150 и 200 эпох. Такой подход позволяет наблюдать динамику изменения производительности модели в зависимости от количества итераций обучения. Анализируя результаты по мере увеличения числа эпох, становится возможным определить оптимальное количество итераций, при котором достигается баланс между достаточной точностью и предотвращением переобучения.

Этап начального обучения (1-50 эпох): на начальном этапе модель активно изучает базовые характеристики классов, что отражается в быстром снижении значения функции потерь и улучшении метрик точности.

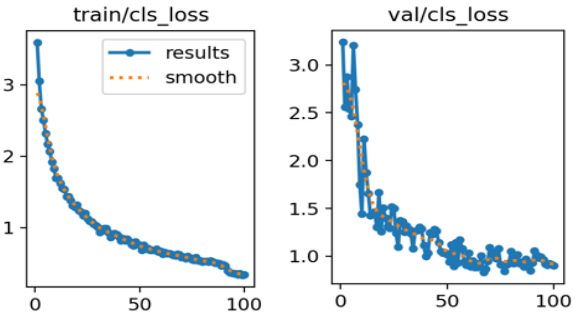
Этап углубленного обучения (50-150 эпох): в этом диапазоне модель продолжает совершенствовать детекцию и классификацию, но темпы улучшения могут несколько замедлиться, указывая на приближение к оптимальной точности.

Этап стабилизации (150-200 эпох): в поздние эпохи следует внимательно отслеживать динамику, чтобы избежать переобучения, когда модель начинает чрезмерно подстраиваться под обучающую выборку, что может негативно сказаться на валидации.

В целях визуализации процесса обучения на рисунках ниже представлены графики изменения значения функции потерь по эпохам. Эти графики наглядно демонстрируют, как функция потерь уменьшается в течение обучения, отражая способности модели эффективно обучаться на предоставленных данных.

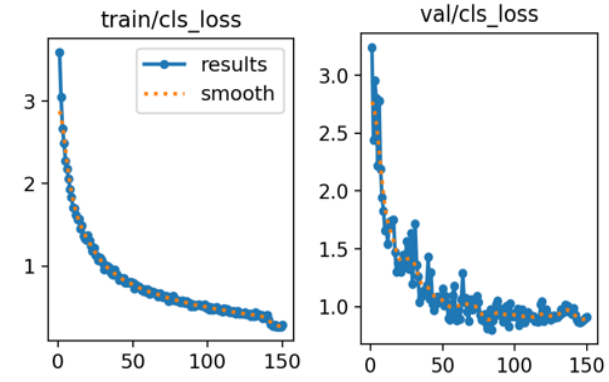


а

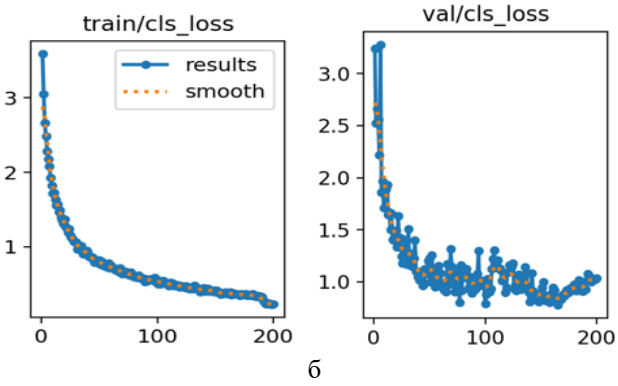


б

Рис. 5 . Кривые обучения для: а) 50 эпох, б) 100 эпох. Графики справа – обучающая выборка, слева – валидационная.



а



б

Рис. 6 . Кривые обучения для: а) 150 эпох, б) 200 эпох. Графики справа – обучающая выборка, слева – валидационная.

Из графиков видно, что на 200-х эпохах начинается переобучение модели (значение функции потерь на валидации начинает увеличиваться). Для 150 же эпох значение функции потерь наименьшее, значит оптимальное количество эпох для обучения – 150.

B. Расчёт метрик на тестовой выборке для YOLOv8

На предыдущем этапе была успешно обучена модель YOLOv8. Следующим шагом является оценка качества её работы, что достигается с помощью таких метрик, как точность (precision), полнота (recall) и F1-мера (F1-score). Они основываются на следующих ключевых понятиях:

- TP (True Positive, истинно положительные)- это количество объектов, которые модель правильно идентифицировала как принадлежащие к определенному классу. В контексте задачи обнаружения животных это означает, что модель правильно обнаружила и классифицировала, например, собаку как собаку.
- FP (False Positive, ложные положительные)- это количество объектов, которые модель ошибочно идентифицировала как принадлежащие к определенному классу. Например, если модель ошибочно определила кошку как собаку, это будет считаться FP для класса "собаки".
- FN (False Negative, ложные отрицательные)- это количество объектов, которые модель не смогла идентифицировать как принадлежащие к определенному классу. Например, если модель не обнаружила собаку на изображении, это будет считаться FN для класса "собаки".

На основе этих понятий введём вышеупомянутые метрики качества:

- Precision (Точность)- эта метрика показывает, какая доля обнаруженных моделью объектов на самом деле является правильной. Она указывает на точность модели в идентификации объектов: чем выше precision, тем меньше ложных срабатываний.

$$Precision = \frac{TP}{TP + FP}$$

- Recall (Полнота)- эта метрика показывает, какая доля всех реальных объектов была правильно обнаружена моделью. Она указывает на способность модели обнаруживать все объекты определенного класса: чем выше recall, тем меньше пропущенных объектов.

$$Recall = \frac{TP}{TP + FN}$$

- F1-score (F1-мера)- это гармоническое среднее между precision и recall, которое используется для баланса этих двух метрик. F1-score особенно полезен, когда важно учитывать как точность, так и полноту модели, например, в случаях, где баланс между FP и FN критичен.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Данные метрики являются критически важными для оценки моделей детекции объектов, поскольку они позволяют сбалансированно оценить, как способность модели правильно идентифицировать объекты, так и её эффективность в обнаружении всех релевантных объектов на изображении. Precision фокусируется на снижении ложных срабатываний, что важно для уменьшения ошибок классификации, тогда как Recall акцентирует внимание на способности модели выявлять максимальное количество объектов, минимизируя пропуски. F1-

score объединяет эти аспекты, предоставляя единое значение, которое учитывает компромисс между точностью и полнотой, что особенно полезно в ситуациях, где необходимо поддерживать баланс между FP и FN. На следующих рисунках приведены метрики модели, рассчитанные на тестовом датасете, состоящем из 100 изображений, а также примеры работы модели и её матрица ошибок.

Таблица 1 – Ключевые метрики модели

Class	Precision	Recall	F1
Все	**0.938**	**0.720**	**0.814**
Барсук	0.889	1.000	0.941
Медведь	1.000	0.699	0.823
Олень	1.000	0.315	0.479
Лиса	0.889	0.769	0.825
Заяц	1.000	0.676	0.807
Ёж	0.987	1.000	0.993
Енот	1.000	0.752	0.858
Белка	0.933	0.700	0.800
Северный Олень	0.681	0.714	0.697

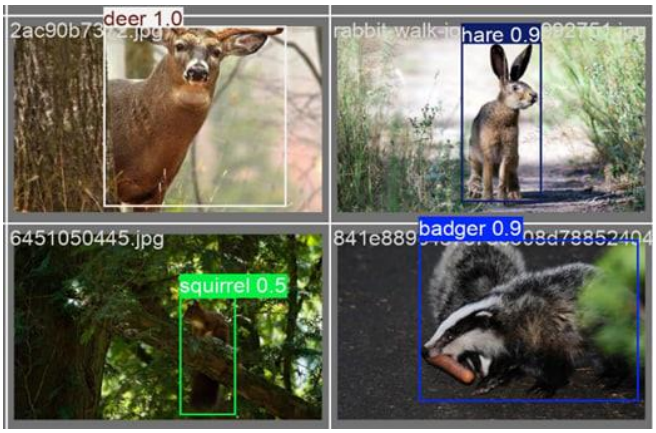


Рис. 7 .Пример работы модели YOLOv8

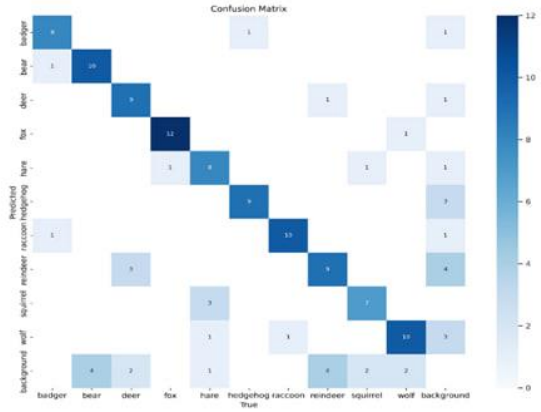


Рис. 8 . Матрица ошибок

C. Обучение модели Faster R-CNN

Обучение модели Faster R-CNN проводилось в 10000 итераций, используя архитектуру Faster R-CNN с ResNet-50 и FPN (Feature Pyramid Network), в соответствии с конфигурацией faster_rcnn_R_50_FPN_3x. Модель была обучена с использованием следующих параметров:

- Максимальное количество итераций: 10000
- Период оценки: 200 итераций
- Базовый learning rate: 0.00025
- Число классов: 10
- Размер пакета на изображение для ROI heads: 128
- Количество рабочих потоков для загрузки данных: 2

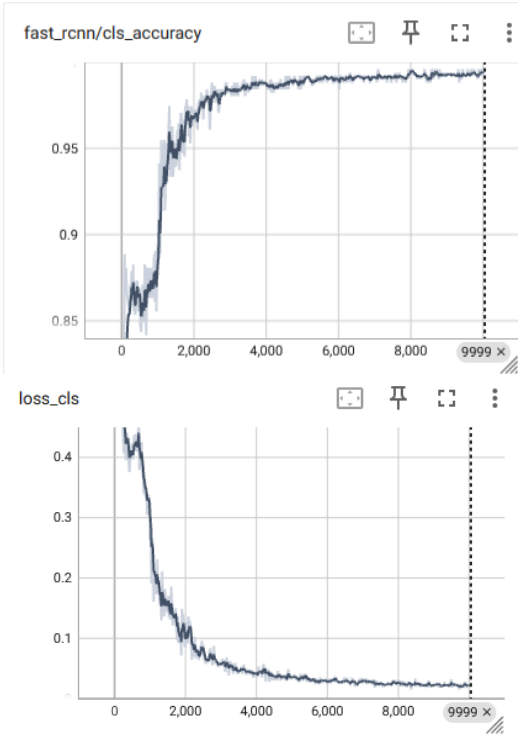


Рис. 9. Кривые Classification accuracy и Classification loss

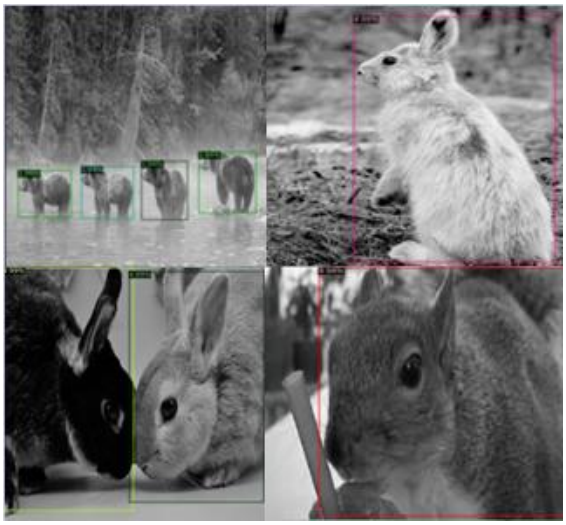


Рис. 10 . Пример работы модели Faster R-CNN

V. Сравнение

Сравним две нейросети, YOLOv8 и Faster R-CNN, рассмотрев изображения из тестовой выборки. Рассмотрим на примере фотографии ежа

Рис. 11. Пример фотографии ежа, используемой для сравнения

Рис. 12 . Пример фотографии, полученный после обработки Faster R-CNN

Рис. 13. Пример фотографии, полученный после обработки YOLOv8

VI. заключение

В ходе исследования были подробно рассмотрены и оценены модели YOLOv8 и Faster R-CNN для задачи детекции объектов, в частности, для обнаружения и классификации животных на изображениях. Модели обучались и тестировались на специально подготовленных наборах данных, что позволило глубже понять возможности и ограничения данной архитектуры.

Таблица 3. Метрики YOLOv8 и Faster R-CNN

Класс	Модель	Эпохи	Precision	Recall	F1-score
All	YOLO	50	0.938	0.720	0.814
	Faster R-CNN	50	0.950	0.650	0.780
	YOLO	100	0.950	0.750	0.838
	Faster R-CNN	100	0.960	0.720	0.830
	YOLO	150	0.960	0.770	0.855
	Faster R-CNN	150	0.965	0.750	0.845

В статье была проведена оценка моделей YOLOv8 и Faster R-CNN с точки зрения их архитектуры и производительности на наборе данных, включающем изображения различных классов животных. Основное внимание уделялось качеству классификации объектов и точности детекции. Результаты экспериментов показали, что YOLOv8 продемонстрировала высокую точность и эффективность, особенно в условиях реального времени, что делает её перспективным инструментом для приложений с требованиями к скорости. В то же время Faster R-CNN показала более высокую стабильность и точность на более поздних этапах обучения, что подтверждает её преимущества для более сложных задач с акцентом на высокую точность.

Важно отметить, что данное исследование фокусировалось на сравнении этих двух моделей, что позволяет сделать выводы об их характеристиках и потенциале в контексте задач детекции объектов. Для более глубоких сравнений и анализа других моделей потребуется проведение дополнительных экспериментов с различными архитектурами и наборами данных. Тем не менее, полученные результаты подчеркивают, что обе модели имеют свои сильные стороны и могут быть использованы в зависимости от специфики задачи, будь то задачи с требованием к скорости, как в случае с YOLOv8, или задачи, где критична точность, как в случае с Faster R-CNN.

Литература

[1] Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25), E5716-E5725. DOI: 10.1073/pnas.1719367115

[2] Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.

[3] Ян Эрим Солек. Программирование компьютерного зрения на языке Python / пер. с англ. Слинкн А. А. - М.: ДМК Пресс, 2016 - 312 с.: ил.

[4] Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб. : Питер, 2018. – 480 с. : ил. – ISBN 978-5-496-02536-2.

[5] Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.. : Пер. с англ. - М. : Издательский дом ‘‘Вильямс’’, 2007. - 1408 с.

[6] Макшанов, А.В. Технологии интеллектуального анализа данных: Учебное пособие / А.В. Макшанов, А.Е. Журавлев. - СПб.: Лань, 2018. - 212 с.

[7] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N. Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," 2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[8] Ali, B., Sadekov, R.N. & Tsodokova, V.V. A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems. Gyroscopy Navig. 13, 241–252 (2022). https://doi.org/10.1134/S2075108722040022

[9] Chernyshova, Yulia & Savelyev, B & Solodov, S & Pronichkin, S. (2022). Applying distributed ledger technologies in megacities to face anthropogenic burden challenges. IOP Conference Series: Earth and Environmental Science. 1069. 012028. 10.1088/1755-1315/1069/1/012028.

[10] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-

[11] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-

[12] А. А. Абакумов, В. О. Хуако (2024), Определение положения тела человека с использованием нейронных сетей., https://elibrary.ru/item.asp?id=72973834&pff=1

[13] Карякин А. В. (2024), Исследование задачи детектирования человека с помощью компьютерного зрения, https://elibrary.ru/item.asp?id=72973870&pff=1

[14] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-

ГРНТИ 28.23.37, 34.33.19
ASJC 1703, 1707
OECD 1.02 EP

Применение нейронных сетей в задачах классификации насекомых

С. Д. Овчаренко

кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2409404@edu.misis.ru

***Аннотация* — в данной работе представлено исследование применения предобученной свёрточной нейронной сети на базе архитектуры ResNet-50 и ее дообучение для задачи классификации насекомых. Цель исследования — разработка нейросети для классификации видов насекомых, которая может быть использована в сельском хозяйстве, экологии и биологии. В рамках исследования анализируются две нейросетевые модели, такие как Resnet и DenseNet. Проводится сравнительный анализ их производительности и точности, а также проверка на собственном датасете, что позволяет выявить преимущества и недостатки каждой модели.**

Результаты экспериментов продемонстрировали высокую точность классификации. Разработанная система может быть использована для автоматического мониторинга насекомых в полевых условиях, а также для улучшения методов управления вредителями в сельском хозяйстве.

Ключевые слова — ResNet-50, DenseNet, сверточная нейронная сеть, классификация насекомых, fine-tuning, сельское хозяйство, мониторинг насекомых, компьютерное зрение, глубокое обучение.

I. Введение

Распознавание насекомых — одна из ключевых задач в биоинформатике и экологии, которая способствует мониторингу биоразнообразия, контролю вредителей и изучению экосистем. Традиционные методы, такие как визуальная идентификация или использование ловушек, требуют значительных временных и человеческих ресурсов. В последние годы нейронные сети стали мощным инструментом для автоматизации и повышения точности процессов классификации насекомых [1,2].

Актуальность классификации насекомых обусловлена их значительной ролью в экосистемах и хозяйственной деятельности человека [3,4]. Насекомые играют ключевую роль в опылении растений, переработке органического вещества и являются частью цепей питания. Однако многие виды насекомых также представляют угрозу в качестве вредителей сельскохозяйственных культур, переносчиков болезней и разрушителей инфраструктуры.

С развитием методов глубокого обучения появились новые возможности для решения задачи классификации насекомых. Одной из наиболее известных архитектур является ResNet (Residual Networks), которая благодаря остаточным соединениям решает проблему деградации градиентов в глубоких сетях [5]. Эти сети эффективно работают на задачах классификации, детекции объектов и сегментации.

Альтернативным подходом являются сети DenseNet (Dense Convolutional Networks), которые минимизируют дублирование вычислений, обеспечивая более плотные соединения между слоями [6,7]. DenseNet позволяют эффективно использовать параметры модели, что делает их подходящими для работы с меньшими наборами данных.

Использование нейронных сетей в задаче классификации насекомых открывает новые горизонты для автоматизации мониторинга и управления биоразнообразием. Благодаря обучению на пользовательских наборах данных и подходу fine-tuning, предобученные сети, такие как ResNet-50 и DenseNet, могут быть адаптированы для идентификации конкретных видов насекомых. Эти модели уже продемонстрировали свою эффективность в задачах сельского хозяйства и экологического мониторинга в глубоком обучении [8].

Настоящая работа направлена на исследование применения современных архитектур нейронных сетей для задачи классификации насекомых с применением технологий компьютерного зрения [9]. Анализируются преимущества архитектуры ResNet-50 и DenseNet, их точность, скорость обучения и способность обрабатывать данные с различными условиями съёмки.

II. Наборы данных

A. ImageNet ImageNet

ImageNet ImageNet [10,11] — это обширный датасет, предназначенный для использования в задачах компьютерного зрения, особенно в классификации изображений. Он содержит более 14 миллионов аннотированных изображений, организованных по примерно 22 тысячам категорий. Каждое изображение в ImageNet классифицировано и отмечено согласно категории объекта, который оно изображает, что делает его одним из самых масштабных и разнообразных наборов данных в области искусственного интеллекта.

ImageNet широко используется для обучения сверточных нейронных сетей (CNN) с нуля. Эти сети могут распознавать и классифицировать тысячи

различных объектов благодаря обширному и разнообразному набору изображений. Модели, предварительно обученные на ImageNet, часто используются как основа для дальнейшего обучения на других, менее масштабных или более специализированных датасетах [12]. Перенос обучения позволяет значительно ускорить процесс обучения и улучшить производительность моделей на конкретных задачах.

B. Дополненный датасет

Для обучения и тестирования моделей был использован пользовательский набор данных, включающий изображения различных видов насекомых. Данный набор является дополненным, состоящим из более 50 видов насекомых: «Тараканы», «Мухи», «Богомолы», «Стрекозы», «Цикады», «Клопы», «Осы», «Пчелы», «Шмели», «Шершни» .

Данные изображения включали сложные фоны (например, растительность, почву, листья) различные углы обзора и освещение. Это позволило улучшить обобщающую способность моделей и увеличить их применимость в реальных задачах.

Характеристики набора данных:

1. Структура: данные структурированы в виде иерархии, где:

- верхний уровень — это классы насекомых.
- Каждый класс содержит изображения, сделанные в различных условиях освещения, фона и позиций.

2. Объем: датасет включает несколько сотен изображений, что обеспечило достаточное разнообразие для обучения и тестирования.

3. Формат изображений: все изображения были приведены к формату RGB и размеру 224x224 пикселей, что соответствует входным требованиям для ResNet-50 и DenseNet.

4. Аугментация: для улучшения качества обучения и повышения обобщающей способности моделей применялись следующие методы аугментации:

- Случайное горизонтальное отражение.
- Случайное изменение яркости, контрастности и насыщенности.
- Нормализация с использованием средних и стандартных отклонений, соответствующих обучению на ImageNet.

5. Разделение данных:

- тренировочный набор - 70% изображений,
- тестовый набор - 20% изображений,
- валидационный набор - 10% изображений.

Набор данных состоит из различных категорий, каждая из которых представляет отдельный вид насекомых (рисунок 1).

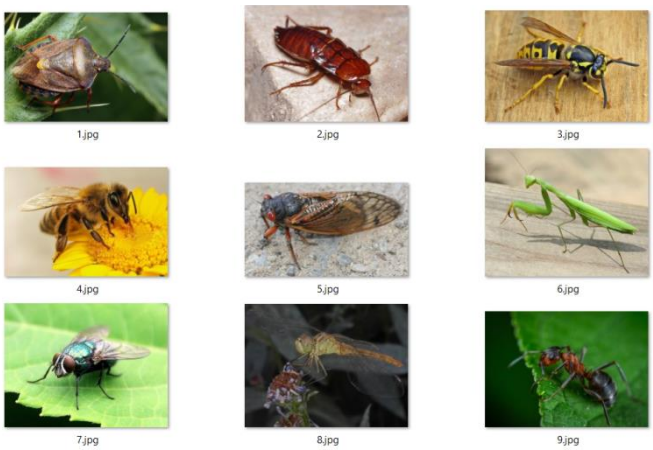


Рис. 1. Примеры кадров различных видов насекомых

Каждое изображение в датасете представлено в формате JPEG и сопровождается аннотацией, содержащей информацию о виде насекомого, его идентификаторе и различных атрибутах (рисунок 2). Набор данных обеспечивает разнообразие изображений, сделанных в различных условиях. Также насекомые запечатлены в различных позах, что позволяет эффективно обучать модели [13].

Рис. 2. Примеры классов насекомых в наборе данных

Каждый вид насекомого обладает уникальными внешними особенностями. Например, некоторые виды жуков и бабочек могут иметь схожие размеры и окраску, что делает их визуально похожими на первый взгляд. Это создаёт сложности в точной классификации. Однако использование современных архитектур, таких как ResNet-50 и DenseNet , позволяет выделять мелкие признаки, которые сложно обнаружить при традиционном подходе. На примере 3-х видов насекомых видно, как разные классы могут быть визуально схожи, но принадлежать к разным категориям, или наоборот — внешне различаться, оставаясь в одном классе. Это подчёркивает важность глубокого анализа данных для точной классификации.

III. Нейросетевые архитектуры

В рамках данной работы основное внимание уделяется современным архитектурам нейронных сетей, таким как ResNet и DenseNet, которые зарекомендовали себя как мощные инструменты для решения задач классификации изображений [14]. Данные архитектуры были выбраны благодаря их высокой точности, эффективности и способности адаптироваться к новым задачам через подход fine-tuning. Особое значение

имеет их применение в задаче классификации насекомых, где требуется обработка изображений с мелкими текстурами и сложными фонами.

Современные архитектуры нейронных сетей играют ключевую роль в решении задач обработки изображений, включая классификацию, детекцию объектов и сегментацию [15]. Среди наиболее успешных подходов выделяются Residual Neural Networks (ResNet) и Dense Convolutional Networks (DenseNet), которые эффективно решают проблемы глубоких сетей, такие как деградация градиентов и чрезмерное дублирование вычислений. Эти архитектуры были выбраны для данного исследования благодаря их высокой точности, гибкости и адаптивности к задачам классификации насекомых. Ниже приводится описание каждой из архитектур и их особенностей:

A. Resnet-50

ResNet-50 является предобученной моделью, изначально обученной на ImageNet, включающем более 1,2 миллиона изображений и 1000 классов. Она использует остаточные соединения, что позволяет обрабатывать глубокие архитектуры без деградации градиента [16,17]. В данной работе ResNet-50 была адаптирована под задачу классификации насекомых с помощью fine-tuning: последний слой сети заменён линейным слоем, настроенным на количество классов в наборе данных. Данная модель хорошо справляется с извлечением сложных признаков изображений, что делает её особенно полезной для анализа насекомых с детализированными текстурами и разнообразными условиями съёмки.

Это сверточная нейронная сеть, разработанная для работы с изображениями, которая известна своей архитектурой, глубиной и эффективностью в решении задач классификации.

Основные особенности ResNet-50:

- Решение проблемы деградации точности в глубоких нейронных сетях. При увеличении количества слоев в обычных нейросетях точность на тестовых данных начинает снижаться из-за градиентного затухания или взрыва, что мешает эффективно обучать глубокие сети.
- Введение остаточных соединений (skip connections), которые позволяют информации проходить через сеть, минуя несколько слоев, и тем самым решают проблему деградации.
- Остаточные соединения: в традиционной нейросети выход каждого слоя передается следующему. В ResNet-50 на выход каждого блока добавляется вход, образуя остаточное соединение:

$$F(x) = x + H(x),$$

где F(x) – итоговая функция, x – вход, а H(x) – выход промежуточных слоев.

Это позволяет сети легче обучаться, так как остаточная связь сохраняет информацию, необходимую для восстановления градиентов.

- Глубина: ResNet-50 состоит из 50 слоев: сверточные слои, объединения (pooling), активации ReLU и остаточные блоки (Residual Blocks).Данная архитектура является "глубокой" версией, более подходящей для задач с большими наборами данных.
- Предобученные веса: ResNet-50 обучается на крупномасштабных наборах данных, таких как ImageNet (более 1 миллиона изображений и 1000 классов). Это делает модель универсальной для различных задач классификации и распознавания.

ResNet-50 стала революцией в компьютерном зрении, и ее применение в биологии, включая распознавание насекомых, демонстрирует ее универсальность и мощь [10]. Эта модель позволяет быстро и точно решать задачи, требующие анализа изображений, что делает ее идеальной для мониторинга и классификации насекомых.

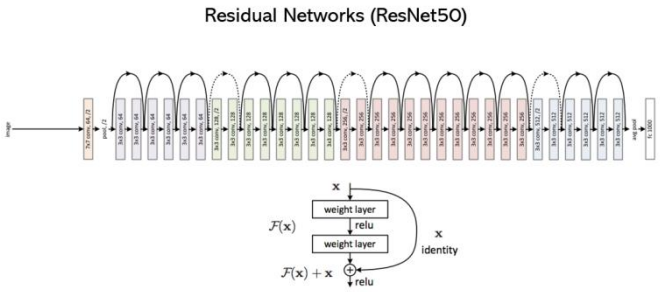


Рис. 3. Архитектура модели ResNet50

- Общая структура сети: входное изображение проходит через начальный сверточный слой с ядром 7×7 и 64 фильтрами, за которым следует операция пулинга (pooling). Сеть состоит из нескольких остаточных блоков, которые объединяют сверточные слои 3×3 и 1×1 с остаточными соединениями. Каждый блок увеличивает количество фильтров (64, 128, 256, 512), уменьшая пространственные размеры изображения через операции пулинга.
- Остаточные соединения показаны дугами между слоями. Эти соединения позволяют пропускать входной сигнал через блок, что помогает избежать деградации градиентов.
- Выходные слои. После сверточных блоков идет операция глобального усреднённого пулинга соответствующими количеству классов в наборе данных ImageNet.
- Пример структуры Residual Block. Входной сигнал (x) подается через два сверточных слоя с функцией активации ReLU. Выход слоя (F(x)) суммируется с оригинальным.

Архитектура позволяет легко изменять глубину сети, добавляя больше блоков. ResNet-50 — это

универсальная архитектура, которая используется для множества задач компьютерного зрения, включая классификацию, детекцию и сегментацию объектов.

B. DenseNet

DenseNet характеризуется плотными соединениями между слоями, где каждый слой получает доступ ко всем предыдущим. В работе DenseNet была использована как альтернатива ResNet-50, демонстрируя схожую точность при меньшем количестве параметров и более эффективном использовании ресурсов. Такая архитектура особенно полезна для работы с ограниченными наборами данных, обеспечивая высокую обобщающую способность [18].

Архитектурные компоненты DenseNet:

- Dense Block (Плотный блок) - основной элемент архитектуры. Каждый слой внутри блока получает входы от всех предыдущих слоёв и передаёт свои выходы всем последующим. Это достигается через конкатенацию выходов. Если имеется несколько слоёв, то каждый слой получает l·k входов, где k— ростовой коэффициент (*growth rate*), который определяет количество новых признаков, добавляемых каждым слоем.
- Transition Layers (Переходные слои)- разделяют плотные блоки. Выполняют уменьшение размерности карты признаков через операции 1×1 свёртки и пулинга, чтобы сократить вычислительные затраты и размерность данных.
- Growth Rate (Коэффициент роста) - указывает, сколько новых признаков добавляет каждый слой. Типичные значения k=12,24,32, где большее значение k увеличивает сложность модели.
- Global Average Pooling. После всех плотных блоков используется операция глобального усреднённого пулинга, которая сокращает размерность данных перед линейным классификатором.
- Output Layer (Выходной слой) - линейный слой для классификации. В случае использования DenseNet на ImageNet он включает 1000 классов.

DenseNet является мощным инструментом для задач, где важна как высокая точность, так и эффективность использования ресурсов [19].

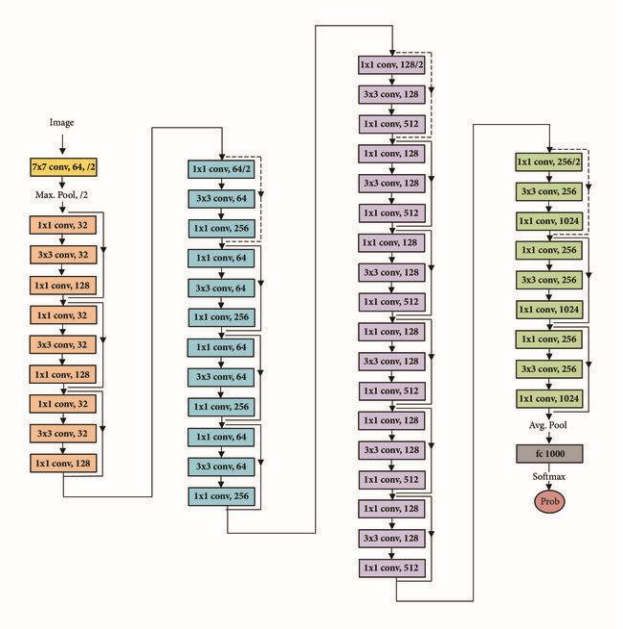


Рис. 4. Архитектура модели DenseNet

Описание архитектуры:

- Входной слой: Принимает изображение и передаёт его в первый свёрточный слой.
- Плотные блоки (Dense Blocks) .Каждый блок состоит из нескольких слоёв, где каждый слой получает на вход все предыдущие слои блока. Такое соединение обеспечивает эффективную передачу информации и повторное использование признаков.
- Переходные слои (Transition Layers). Разделяют плотные блоки и выполняют операции свёртки и пулинга для уменьшения размерности данных.
- Выходной слой . После последнего плотного блока применяется глобальный усреднённый пулинг, затем следует полностью связанный слой для классификации.

Такая структура позволяет DenseNet эффективно использовать параметры модели, улучшать передачу градиентов и достигать высокой точности в задачах классификации изображений.

IV. Сравнение результатов

Сравнение этих двух архитектур ResNet50 и DenseNet показало, что DenseNet требует меньше параметров, чем ResNet, и имеет лучшую обобщающую способность при работе с небольшими наборами данных. Однако ResNet-50 превосходит DenseNet по скорости предсказания и является более устойчивой к увеличению сложности данных. Оба подхода дополняют друг друга: ResNet-50 идеально подходит для задач с большими наборами данных, тогда как DenseNet демонстрирует превосходство в условиях ограниченных ресурсов или ограниченного объёма данных [20].

Использование этих архитектур в данной работе показало, что обе сети эффективно справляются с задачей классификации насекомых. ResNet-50 хорошо зарекомендовала себя благодаря своей стабильности и высокой точности, а DenseNet продемонстрировала отличную способность выделять признаки даже в сложных условиях съёмки.

При сравнении производительности ResNet50 и DenseNet вступают несколько факторов:

- Точность : обе модели демонстрируют высокую точность на эталонных наборах данных, таких как CIFAR-10 и ImageNet. Однако DenseNet часто превосходит ResNet по точности благодаря эффективному повторному использованию функций и лучшему градиентному потоку [21].

Набор данных случайным образом разделен на 80% обучающей и 20% проверочной выборки, чтобы оценить эффективность модели.

A. *Процесс обучения модели ResNet-50*

По результатам обучения на протяжении 50 эпох модель продемонстрировала следующие показатели:

1. Accuracy на тренировочной выборке -0,9784;

2. Loss на тренировочной выборке - 0,1201;

3. Accuracy на валидационной выборке - 0,9721.

4. Loss на валидационной выборке - 0,0132;

На рисунках 4-5 показаны точность и потери реализованных моделей ResNet-50.

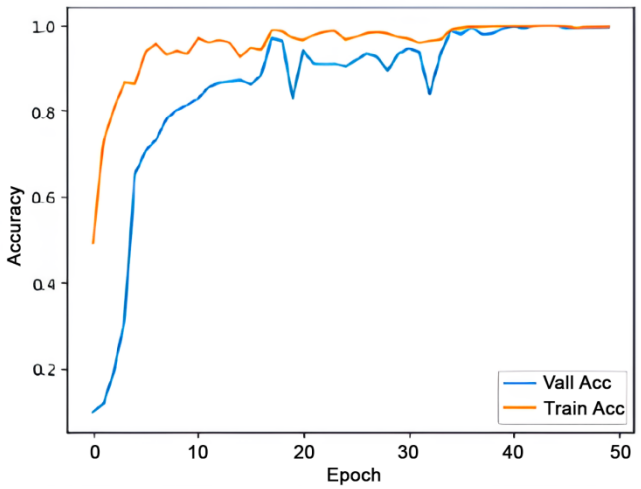


Рис. 4. Результаты работы модели ResNet50 (Accuracy)

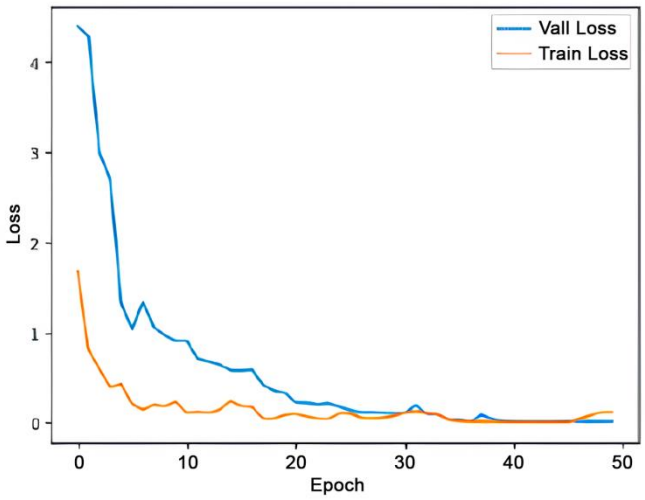


Рис. 5. Результаты работы модели ResNet50 (Loss)

B. *Процесс обучения модели DenseNet*

По результатам обучения на протяжении 50 эпох модель продемонстрировала следующие показатели:

1. Accuracy на тренировочной выборке - 0,7044;

2. Loss на тренировочной выборке - 0,1344;

3. Accuracy на валидационной выборке - 0,8844.

4. Loss на валидационной выборке - 0,0110;

На рисунках 4-5 показаны точность и потери реализованных моделей ResNet-50.

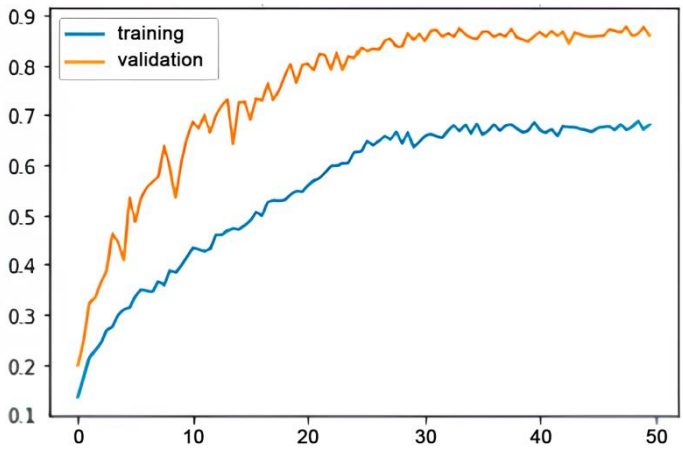


Рис. 6. Результаты работы модели DenseNet (Accuracy)

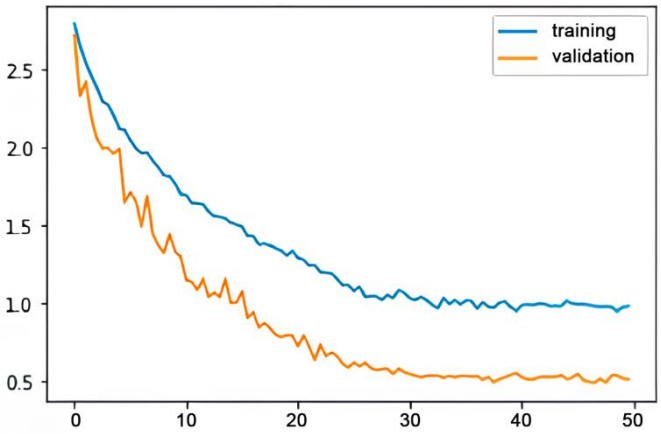


Рис. 7. Результаты работы модели DenseNet (Loss)

Вычислительная эффективность: в то время как ResNet50 эффективен с точки зрения вычислений, архитектура DenseNet позволяет достичь аналогичной или более высокой производительности при меньшем количестве параметров, что делает его отличным кандидатом для среды с ограниченными ресурсами. Данные, приведенные в таблице 1, отображают количественные оценки для двух подходов.

Таблица 1. Показатели производительности точности

Model	Accuracy
ResNet-50	0,9721
DenseNet-121	0,8844

V. Заключение

В данной работе была рассмотрена задача классификации насекомых, используя современные архитектуры глубоких нейронных сетей ResNet-50 и DenseNet. В рамках исследования была проведена предобработка данных, включая аугментацию изображений, что позволило повысить устойчивость моделей к шуму и разнообразию фонов. Сравнительный анализ производительности двух моделей показал, что Resnet-50 обладает преимуществами в точности классификации на сложных изображениях, тогда как DenseNet демонстрирует высокую эффективность при обработке больших объёмов данных благодаря своей остаточной архитектуре.

Результаты экспериментов подтверждают, что глубокие нейронные сети могут быть успешно применены для классификации насекомых в полевых условиях. Модели обеспечивают высокую точность, что делает их подходящими для реальных задач, таких как сельское хозяйство и мониторинг биоразнообразия в полевых условиях.

Литература

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of CVPR. DOI: 10.1109/CVPR.2016.90

[2] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Proceedings of CVPR. DOI: 10.1109/CVPR.2017.243

[3] Ступина, А. А. Исследование возможности распознавания животных в искусственной среде / А. А. Ступина // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : СБОРНИК СТАТЕЙ НАУЧНО-ТЕХНИЧЕСКОГО СЕМИНАРА СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ», Москва, 30 декабря 2023 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2023. – С. 62-67. – EDN HRZEPC

[4] Антипов, И. И. Исследование возможности определения возраста клиента при помощи компьютерного зрения / И. И. Антипов // Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях : сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики", Москва, 30–31 мая 2024 года. – Москва: Национальный исследовательский технологический университет "МИСИС", 2024. – С. 12-16. – EDN VDLJDP.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. CVPR. DOI: 10.1109/CVPR.2016.91

[6] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal Loss for Dense Object Detection. ICCV. DOI: 10.1109/ICCV.2017.324

[7] Reddy, P. P., & Chaudhuri, S. R. (2021). Automated Pest Detection and Classification Using Deep Learning Techniques: A Review. Artificial Intelligence in Agriculture. DOI: 10.1016/j.aiia.2021.03.001

[8] Бикмаев, Р. Р. Особенности применения свёрточных нейронных сетей в задаче распознавания морских надводных объектов / Р. Р. Бикмаев, Р. Н. Садеков // Известия Института инженерной физики. – 2019. – № 4(54). – С. 105-110. – EDN FBYJMA.

[9] Martínez, G., Larrañaga, A., & Jiménez, A. (2020). Automatic Insect Detection in Crops Using Deep Neural Networks. Computers and Electronics in Agriculture, 174, 105515. DOI: 10.1016/j.compag.2020.105515

[10] Shorten, C., & Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 60. DOI: 10.1186/s40537-019-0197-0

[11] Система технического зрения как источник дополнительной информации в задаче автомобильной навигации / С. Б. Беркович, Н. И. Котов, А. В. Лычагов [и др.] // Гироскопия и навигация. – 2017. – Т. 25, № 1(96). – С. 49-63. – DOI 10.17285/0869-7035.2017.25.1.049-063. – EDN YKGWIJ.

[12] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision. DOI: 10.1007/s11263-015-0816-y

[13] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint.

[14] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. ICLR.

[15] Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. CVPR. DOI: 10.1109/CVPR.2016.282

[16] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ICML. DOI: 10.48550/arXiv.1905.11946

[17] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks? NeurIPS. DOI: 10.48550/arXiv.1411.1792

[18] Abadi, M., Agarwal, A., Barham, P., et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI.

[19] Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. NeurIPS. DOI: 10.48550/arXiv.1912.01703

[20] Szegedy, C., Liu, W., Jia, Y., et al. (2015). Going Deeper with Convolutions. CVPR. DOI: 10.1109/CVPR.2015.7298594

[21] Berman, M., Triki, A. R., & Blaschko, M. B. (2018). The Lovász-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks.

ГРНТИ 34.33.07, 68.41.29, 68.41.27
ASJC 1707, 1706, 1702, 2209,1102
OECD 01.03.AG, 02.08.SR, 01.06.BA

Классификация болезней томатов при помощи компьютерного зрения

А. Р. Панкратов
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m2403748@edu.misis.ru

Т. В. Конев
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m1911179@edu.misis.ru

***Аннотация* — в современном сельском хозяйстве важным аспектом является своевременная диагностика заболеваний растений. В данной работе рассматривается применение методов компьютерного зрения и глубокого обучения для автоматической классификации заболеваний листьев томатов. Основная цель — разработка системы, основанной на архитектуре ResNet, для классификации изображений на здоровые листья и листья с различными заболеваниями. Результаты исследования демонстрируют высокую точность модели и подчеркивают её практическую значимость для фермеров и агрономов.**

Ключевые слова — обработка изображений, классификация, глубокое обучение, ResNet, компьютерное зрение, агротехнологии.

I. Введение

Болезни растений представляют одну из самых серьёзных угроз для сельского хозяйства, приводя к значительным экономическим потерям и снижению качества продукции. Томаты, как одна из ключевых сельскохозяйственных культур [1], подвержены множеству заболеваний, таких как фитофтора, мучнистая роса и пятнистость. Эти заболевания могут быстро распространяться, если не предпринять своевременных мер, это делает диагностику критически важной.

Традиционные методы диагностики, основанные на визуальном осмотре и консультации специалистов, имеют ряд недостатков. Во-первых, они требуют значительных временных и человеческих ресурсов. Во-вторых, результаты сильно зависят от опыта эксперта, что может приводить к ошибкам. Автоматизация процесса диагностики с использованием технологий компьютерного зрения и глубокого обучения способна решить эти проблемы, обеспечивая высокую точность и скорость классификации.

Сверточные нейронные сети (CNN) зарекомендовали себя как мощный инструмент для анализа изображений, особенно в задачах классификации [2]. Однако увеличение глубины сети часто приводит к проблемам, таким как исчезающие градиенты, что затрудняет обучение. Современные архитектуры, такие как ResNet, успешно решают эту проблему за счёт использования остаточных блоков, позволяя эффективно обучать глубокие модели.

Целью данного исследования является разработка интеллектуальной системы, способной классифицировать состояние листьев томатов (здоровые или с различными заболеваниями) на основе анализа изображений. В статье рассматриваются этапы подготовки данных, разработка модели, её обучение и тестирование, а также анализ результатов и потенциальное применение системы в сельском хозяйстве.

II. Наборы данных

Эффективность моделей глубокого обучения в значительной степени определяется качеством данных, на которых они обучаются. В рамках данной работы для обучения и тестирования исследуемых нейронных сетей применялись как собственные локальные наборы данных, собранные авторами, так и общедоступные наборы. В этом разделе подробнее рассмотрим используемые датасеты.

A. PlantVillage Dataset:

- Этот общедоступный датасет [3] включает изображения листьев томатов, разделённые на несколько классов, таких как здоровые листья, фитофтора, мучнистая роса и другие заболевания.
- Общее количество изображений в датасете составляет около 20 000, что обеспечивает широкий охват различных сценариев заболеваний.
- Каждый класс сбалансирован, чтобы избежать смещения в обучении модели.

Рис. 1 - Пример изображений из датасета PlantVillage

B. *Собранные данные в полевых условиях:*

- Для повышения реалистичности модели были добавлены снимки, сделанные в реальных полевых условиях с разным уровнем освещения, углом съёмки и качеством изображений.
- Эти данные позволяют модели адаптироваться к различным условиям эксплуатации в реальных аграрных процессах.
- Особое внимание уделялось расширению представленных заболеваний, включая редкие случаи.

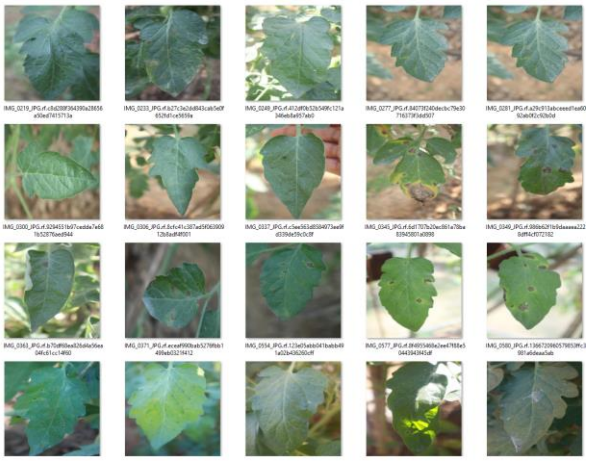


Рис. 2 - Пример полевых снимков листьев с различными уровнями повреждений

Разделение данных:

Для эффективного обучения и оценки модели данные были разделены следующим образом:

- Обучающая выборка (70%) — используется для настройки параметров модели.
- Валидационная выборка (20%) — позволяет оценить производительность модели на промежуточных этапах.
- Тестовая выборка (10%) — используется для окончательной проверки качества модели на новых данных.

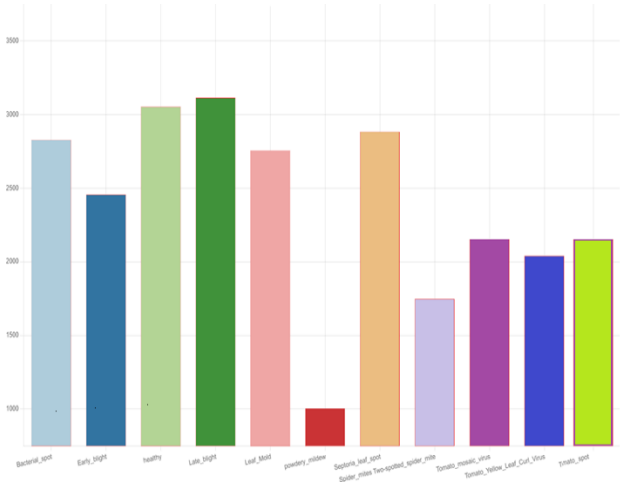


Рис. 3- Распределение данных по классам заболеваний

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

В представленной работе для решения задачи классификации болезней листьев томата использовались две нейронные сети: InceptionV3 и ResNet50. Обе модели представляют собой мощные инструменты для извлечения высокоуровневых признаков [4] из изображений, что важно для точной классификации сложных объектов, таких как листья томатов с различными заболеваниями.

A. Inseption V3

Для классификации 11 типов заболеваний листьев томата была использована сверточная нейросетевая модель. Модель включает в себя несколько сверточных слоев [5], за которыми следуют объединяющие слои и полностью связанные слои. Inceptionv3 — это одна из глубоких сверточных нейронных сетей, разработанная Google для обработки изображений. Она использует инновационные "Inception" блоки, которые позволяют эффективно извлекать различные уровни признаков из входных изображений. Архитектура [6] модели представлена на рисунке 4.

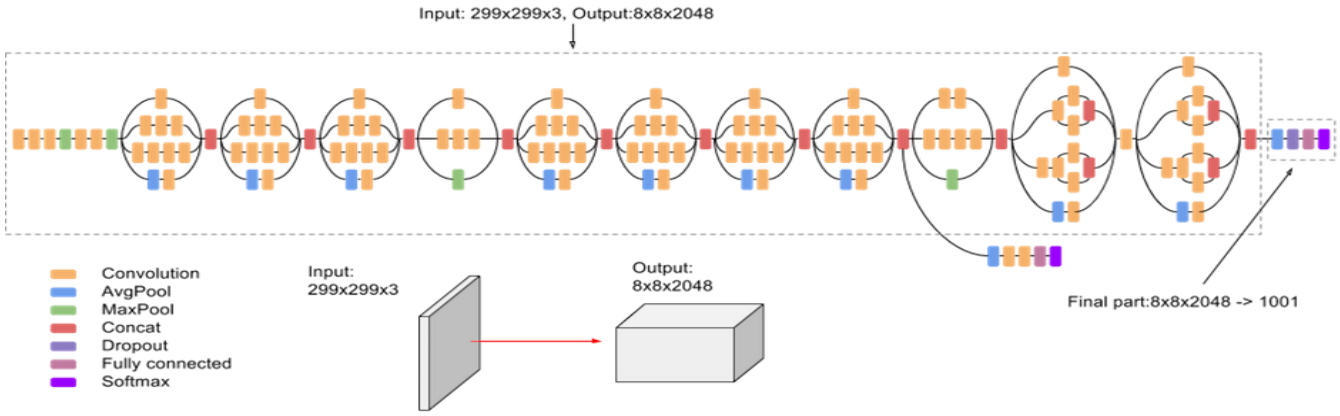


Рис. 4 - Архитектура InceptionV3

Для классификации используются веса предобученной на наборе данных ImageNet [7] модели, доступные в TensorFlow. Модель дообучается на изображениях из набора данных PlantVillage Dataset. Дообучение длится 5 эпох с размером батча 128. Выходной слой модели представляет собой полностью связанный слой с 11 выходными единицами, соответствующими каждому классу заболевания листьев томата. В качестве оптимизатора был выбран оптимизатор Adam с фиксированной скоростью обучения 0,0001. Для предотвращения переобучения применялась регуляризация весов с использованием L2-нормы. В качестве функции потерь используется кросс-энтропия в задаче мульклассовой классификации (cate-gorical crossentropy) – softmax активация в сочетании с кроссэнтропийной функцией потерь. Кросс-энтропия (Crossentropy) [8],

$$C = -\left(\frac{1}{N}\right) * \sum i(y_i * \log(y_i')) \quad (1)$$

Где:

N - количество выборок

y_i - истинная метка для i-й выборки

y'_i - прогнозируемая вероятность для i-й выборки

Также в процесс были включены нормализация и масштабирование, а также Label Encode для кодирования строковых меток классов в числовые индексы, а также аугментация изображений, включающая в себя:

- цветовую коррекцию,
- отражение по горизонтали,
- случайное вращение.

B. Resnet50

ResNet-50 представляет собой глубокую сверточную нейронную сеть, включающую 50 слоёв с обучаемыми весами. Одной из ключевых особенностей архитектуры является использование пропускных соединений (skip connections), которые позволяют эффективно обучать модель, сохраняя важные характеристики изображений и улучшая устойчивость к исчезновению градиентов. Архитектура модели приведена на рисунке 5.

Как и модель Inception V3, архитектура ResNet-50 была дообучена на изображениях из набора данных PlantVillage Dataset. Процесс дообучения включал 5 эпох с размером батча 128. Выходной слой модели классифицировал изображения в один из 11 классов заболеваний, соответствующих классам из представленных датасетов.

Для оптимизации использовался алгоритм Adam с фиксированной скоростью обучения 0,0001. Регуляризация весов выполнялась с использованием L2-нормы. В качестве функции потерь применялась кроссэнтропия (categorical crossentropy) в сочетании с активацией softmax.

Для подготовки данных использовались стандартные процедуры нормализации, масштабирования и кодирования меток (Label Encoding). Дополнительно применялась аугментация данных, направленная на расширение обучающей выборки и повышение устойчивости модели к условиям реального мира, таким как изменения освещения, углы съёмки и другие факторы. Эти методы позволили улучшить общую производительность модели и её адаптацию к реальным задачам.

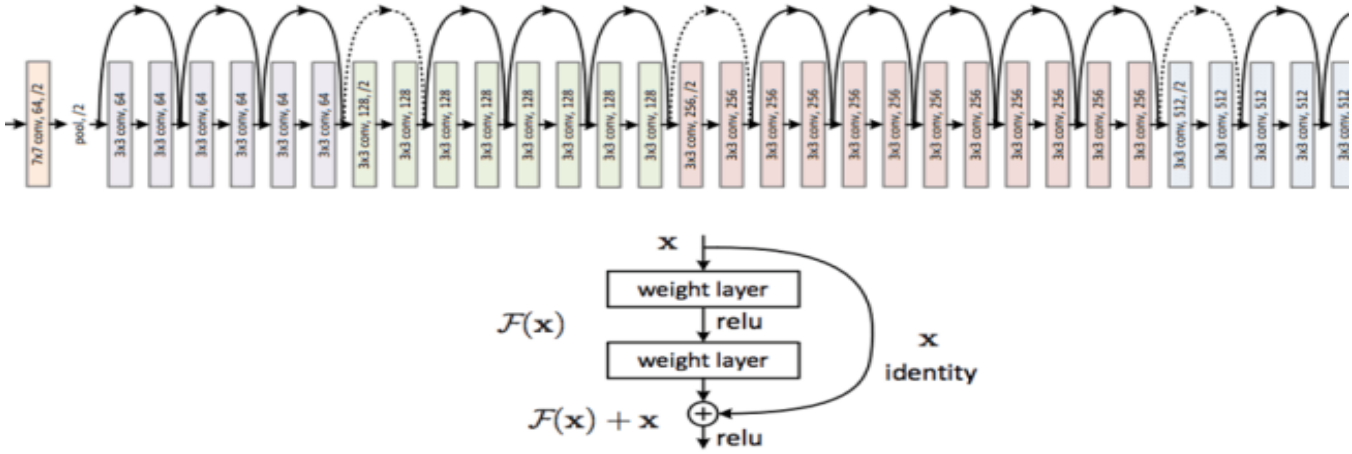


Рис. 5 - Архитектура ResNet

IV. Результаты

Точность модели выступает ключевым индикатором её способности правильно классифицировать заболевания томатных листьев. Для оценки эффективности работы модели использовался показатель F1-score, который позволяет сбалансированно учитывать как полноту (recall), так и точность (precision) классификации. Формула для вычисления F1-score представлена ниже:

$$F1 = 2\,\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \qquad (2)$$

где Precision – точность. Recall – полнота.

В свою очередь

- Precision (точность) рассчитывается как:

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

- Recall (полнота) рассчитывается по формуле:

$$Recall = \frac{TP}{TP + FN} \qquad (4)$$

Разберём обозначения TP, FP, FN, TN
в контексте задачи, связанной с классификацией заболеваний томатных листьев:

- True Positive (TP)- количество случаев, когда заболевание действительно присутствует в выборке, и модель успешно классифицировала его как заражённое.
- False Positive (FP)- количество объектов, которые фактически являются здоровыми, но модель ошибочно отнесла их к заражённым.
- False Negative (FN)- количество случаев, когда заболевание действительно присутствует, но модель не смогла его идентифицировать.
- True Negative (TN)- количество объектов, которые на самом деле являются здоровыми, и модель корректно классифицировала их как таковые.F1_score для каждой модели отображена в таблице:

Таблица 1. F1_score каждой модели

Модель	F1_score
Inseption V3	0.7117
ResNet50	0.8396

Из таблицы видно, лучшие результаты имеет модель ResNet50. Точность модели составляет 0.8396, что означает, что 83,96% положительных прогнозов, сделанных моделью, были правильными. Это признак того, что модель хорошо работает. Модель Inseption V3 показала результаты значительно хуже по сравнению с моделью ResNet50.

Результаты обучения модели, отражающие различные виды заболеваний, определенные данной моделью, представлены на рисунке 6.

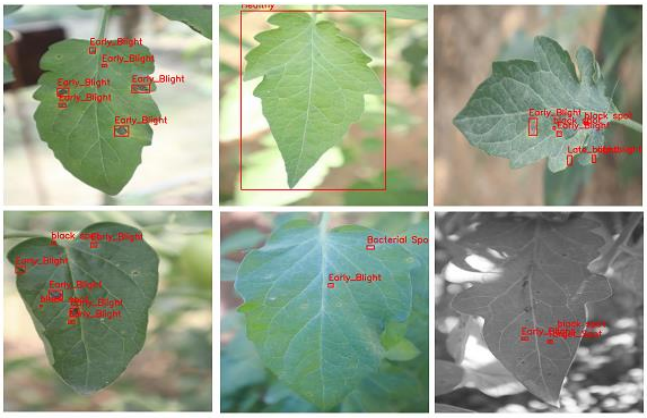


Рис. 6 - Определение заболевания

Рисунок 7 демонстрирует точность модели ResNet50 по каждому классу заболеваний.

Class	Precision	Recall	F1-Score
Bacterial_spot	0.8652	0.8407	0.8528
Early_blight	0.8442	0.8580	0.8511
Late_blight	0.8698	0.8851	0.8774
Leaf_Mold	0.8630	0.8630	0.8630
Septoria_leaf_spot	0.8667	0.8569	0.8618
Spider_mites	0.7717	0.8026	0.7868
Target_Spot	0.8024	0.8167	0.8095
Tomato_Yellow_Leaf_Curl_Virus	0.8214	0.7892	0.8050
Tomato_mosaic_virus	0.8264	0.8083	0.8173
Healthy	0.8886	0.8893	0.8890
Powdery_mildew	0.6381	0.6634	0.6505
Accuracy			**0.8396**

Рис. 7 - Метрики классификации

После завершения обучения модель была протестирована на тестовом наборе данных (рисунок 8). Результаты показали, что точность модели достигла 83,4%, что подтверждает её высокую способность корректно классифицировать заболевания томатов по изображениям листьев. Были выявлены несколько типов заболеваний, для которых модель демонстрировала низкую точность, что указывает направления дальнейшего совершенствования модели.

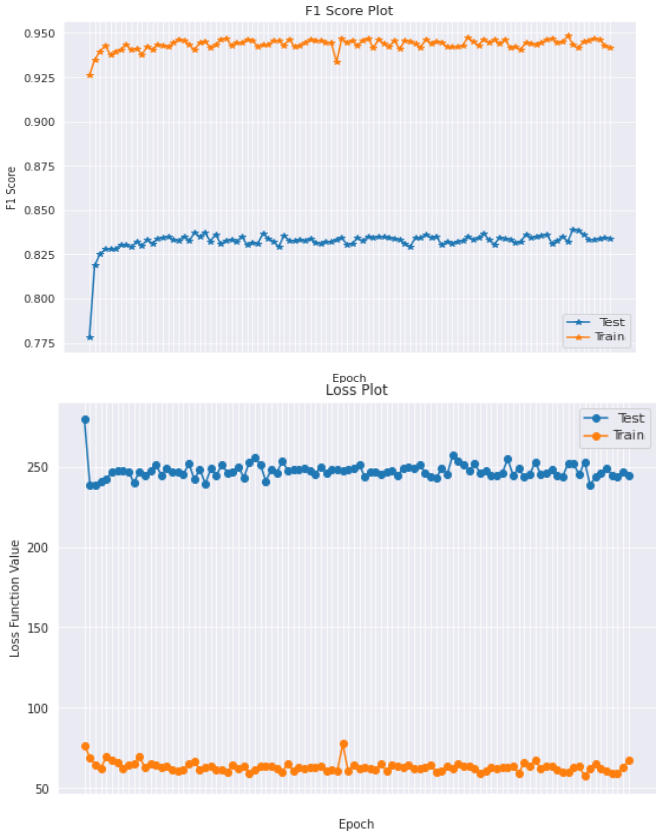


Рис. 8 - График обучения и тестирования модели

Для наглядности результаты тестирования были визуализированы в виде матрицы точности (рисунок 9), которая показывает, как распределяются ошибки классификации между различными классами.

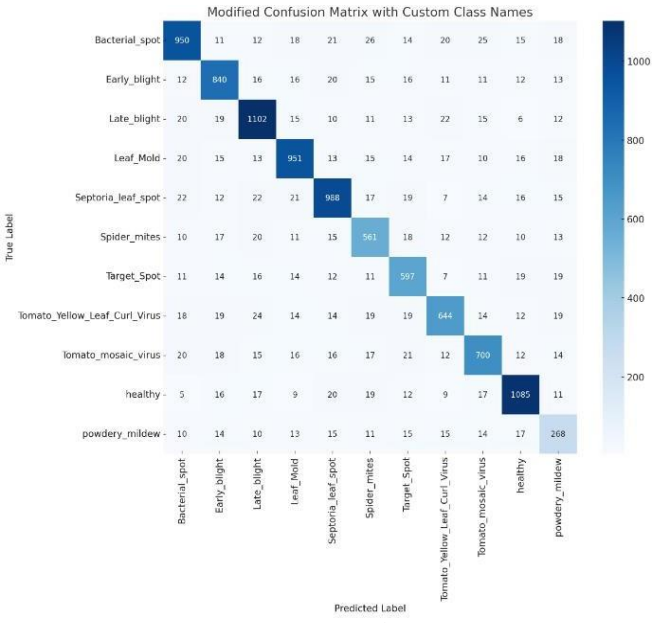


Рис. 9 - Матрица точности обученной модели

V. Заключение

В настоящем исследовании были подробно рассмотрены основные наборы данных, на которых проводилось обучение и тестирование рассматриваемых нейронных сетей. Кроме того, для эффективной обработки информации был создан собственный набор данных, обеспечивающий более глубокий и точный анализ классификации болезней листьев томатов.

В работе представлены две различные нейронные сети, применяемые для решения задачи классификации. Каждая нейронная сеть рассмотрена в контексте ее архитектуры, процесса обучения, а также использованных для обучения и тестирования наборов данных. Это способствует более глубокому пониманию методологии и технических аспектов проведенного исследования.

Каждая из представленных нейронных сетей была подробно проанализирована, а полученные результаты были подвергнуты сравнительному анализу. По полученным данным можно утверждать, что нейронная сеть ResNet50 демонстрирует определенные преимущества по сравнению с альтернативной сетью InceptionV3. Это выражается в более высокой точности в решении задачи классификации болезней.

В целом, результаты исследования подчеркивают не только значимость использования современных нейронных сетей в области распознавания болезней сельских культур, но и важность выбора оптимальной архитектуры для конкретной задачи.

Литература

[1] Микрюков Т.В. Основные угрозы, влияющие на экономическую безопасность сельскохозяйственных организаций. // Вестник Удмуртского университета. Серия «Экономика и право». — 2012. — № 2. —P. —3.

[2] Francis, M., Deisy, C. Disease detection and classification in agricultural plants using convolutional neural networks — A visual understanding. // IEEE 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019. —№ 6. —P. —1063–1068.

[3] https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

[4] R. F. Berriel, A. T. Lopes, A. F. de Souza, and T. Oliveira-Santos, “Deep Learning Based Large-Scale Automatic Satellite Crosswalk Classification,” IEEE Geoscience and Remote Sensing Letters, vol. 14, pp. 1513–1517, Sept 2017.

[5] R. F. Berriel, F. S. Rossi, A. F. de Souza, and T. Oliveira-Santos, “Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A Deep Learning Approach,” Computers & Graphics, vol. 68, pp. 32–42, Nov 2017.

[6] "Rethinking the Inception Architecture for Computer Vision" - Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna (2016)

[7] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255.

[8] "Information Theory, Inference, and Learning Algorithms" - David J.C. MacKay, 2003, pp. 604.

[9] "Deep Residual Learning for Image Recognition" - Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016)

[10] "Ensemble Methods in Data Mining: Improving Accuracy Through Combining

[11] Фомина А. А. Классификация драгоценных камней при помощи компьютерного зрения // Труды кафедры инженерной кибернетики. — Москва: НИТУ «МИСиС», 2023.

[12] Minkin U. I., Panchenko A. V., Shkanaev A. Y., Konovalenko I. A., Putintsev D. N., Sadekov R. N. Computer Vision System: A Tool for Evaluating the Quality of Wheat in a Grain Tank // JSC Cognitive, Moscow; Institute for Systems Analysis FRC CSC RAS, Moscow; MEI «Institute of Engineering Physics», Serphukhov, 2023

Применение больших языковых моделей в рамках голосового управления роботом-манипулятором посредством естественной речи

Я. С. Савельев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
toma@addit.ru

И. А.Рябухин
Faculty of Informating and Statistics
Prague University of Economics and Business
Прага, Чехия
ryai01@vse.cz

Т. А. Синельникова
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
yar21sav@gmail.com

***Аннотация*— в данной работе раскрываются технические особенности по построению программного модуля с голосовым управлением на основе большой языковой модели LLaMA, с помощью которого робот будет способен воспринимать команду на естественном языке и выполнять манипулятивные действия с внешними объектами. Также будут рассмотрены различные классы языковых моделей, в частности описаны отличительные характеристики LLaMA от других похожих моделей и рассмотрены реальные примеры сценариев применения других больших языковых моделей в робототехнике.**

Ключевые слова — искусственный интеллект, LLM, большие языковые модели, голосовое управление, робототехника, промышленные роботы, естественная речь

I. Введение

На сегодняшний день внедрение промышленных роботов на производства является одним из ключевых направлений в рамках развития технологического суверенитета в России. Этому благоприятствует тот факт, что в России применение роботов в промышленности и на производствах достаточно мало, особенно если сравнивать со странами Азии. Поэтому в приоритете стоят гибкость и масштабируемость разрабатываемых решений, чтобы эти инновации были распространены на территорию всей страны, включая регионы [1].

Причины малого количества применения роботов достаточно разрозненные и диверсифицированные, однако хочется обратить внимание на некоторые из них. Одной из таких причин является тот факт, что существующие роботы-манипуляторы имеют достаточно сложную систему управления: в виде отдельной диспетчерской рубки или пульта управления. Такой тип управления имеет ряд очевидных недостатков, таких как малый потенциал к ремонтопригодности и высокая стоимость обслуживания, поскольку такие системы сами по себе являются программно-аппаратными комплексами, требующими дополнительных затрат на логистику и ремонт.

Параллельно с появившимся запросом от промышленных предприятий невозможно оспаривать бум технологических решений с применением искусственного интеллекта. Поэтому особенную популярность приобрели решения с применением больших языковых моделей (Large Language Model, далее LLM). Такие решения реализуют совершенно новый класс сценариев взаимодействия как с программными, так и с программно-аппаратными комплексами.

Появление робота-манипулятора, способного качественно интерпретировать голосовые команды на естественном языке и выполнять команды по манипулированию объектами, потенциально является одним из наиболее востребованных запросов современного рынка, поскольку такая система позволит оптимизировать затраты на ремонт и обслуживание за счет отсутствия логистических затрат и уход от аппаратного управления.

Данная работа посвящена созданию программного модуля для управления роботом-манипулятором на основе обработки естественной речи с помощью модели LLaMA. Выполнять команды по манипулированию объектами робот должен на основе выбранной модели не только быстро и точно, но и с оценкой наиболее валидного решения [2].

II. Большие языковые модели (LLM)

A. *Классификация LLM моделей*

Современные решения на основе LLM занимаются не только обработкой самого текста, но и специализируются на каком-то формате этого текста: голосовые команды, комментарии в социальных сетях, длинные предложения и т. д. Большие языковые модели построены на основе трансформерной архитектуры, фактически трансформирующей получаемые результаты во входные данные с сохранением семантических связей между компонентами контекста.

Наиболее популярные на сегодня LLM модели: BERT и его русифицированный аналог ruBERT, GPT, Focused Transformers. На рис. 1 приведена тенденция к увеличению интереса к различным LLM решениям.

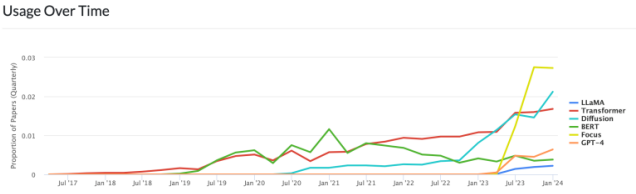


Рис. 1. Количество упоминаний LLM моделей в научных работах

Таким образом, основываясь на количестве упоминаний различных LLM в научных работах, можно отметить впечатляющую динамику роста популярности модели LLaMA, так как в период с осени 2023 года по январь 2024 года динамика количества упоминаний данного класса LLM является максимальной среди других классов.

B. *Модели LLaMA*

Модели класса LLaMA предлагают несколько возможных конфигураций. В рамках исследования была использована модель LLaMA-7B, которая является самой «легковесной» среди всего семейства моделей LLaMA.

В настоящее время областью применения моделей семейства LLaMA является обработка естественного языка, в которой не ожидается текста с высоким количеством профессиональных терминов, интенсивных диалогов и многоязычности. Однако наличие всего необходимого для интеллектуального анализа текста на русском языке делает возможным ее применение в задачи голосового управления роботом-манипулятором. А отсутствие перечисленных выше функциональных возможностей является скорее дополнительным плюсом, так как значительно оптимизирует скорость отклика и место хранения.

Если сравнить модели LLaMA с другими классическими LLM моделями на основе трансформенной архитектуры, то можно увидеть некоторые отличия [3]:

- Наличие пре-нормализации данных. За счет нормализации входных данных для каждого внутреннего слоя модели с помощью RMSNorm подхода происходят улучшения стабильности тренировок. Для сравнения, другие модели используют только нормализацию выходных данных.
- Изменение активационной функции. Вместо ReLU в LLaMA используется SwiGLU, что позволяет значительно улучшить метрики работы.
- Использование поворотно-позиционных вложений (rotary positional embedding или RoPE), которые используются в каждом слое.

На рис. 2 для визуализации отличительных особенностей архитектур моделей представлена архитектура LLaMA и общая архитектура модели-трансформера.

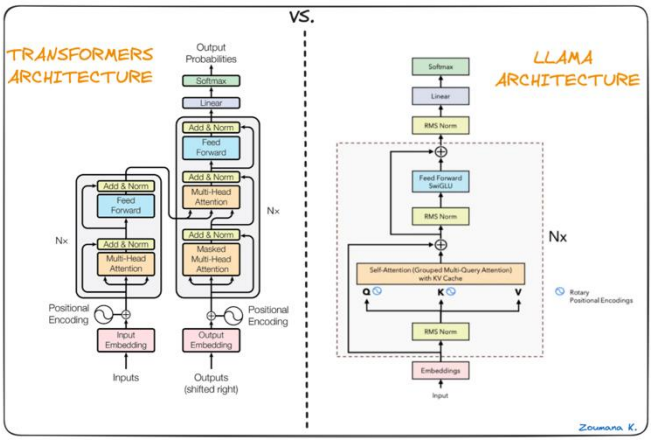


Рис. 2. Сравнение архитектуры классической модели-трансформера и LLaMa [3]

Таким образом, выбранная модель LLaMA-7B готова для использования в промышленной среде в рамках создания и эксплуатации модуля голосового управления робота-манипулятора. А полученное типовое решение является мобильным, широко масштабируемым и стабильно работающим в рамках поставленной задачи и учитывает возможности обязательной синергии в ансамбле различных технологий.

III. Архитектура решения

A. *Общая архитектура робота-манипулятора*

Рассмотрим модульную архитектуру полученного решения. Вначале (см. рис.3) голосовая управляющая команда обрабатывается и переводится в текстовый формат, для этого используется компонент SaluteSpeech от компании Сбер. Он адаптирован под русский язык и работает с подавлением шумов для определения текста. Это особенно важно в поставленной задаче, поскольку использование робота будет происходить в промышленных условиях, а значит ожидаются значительные мало прогнозируемые звуковые помехи.

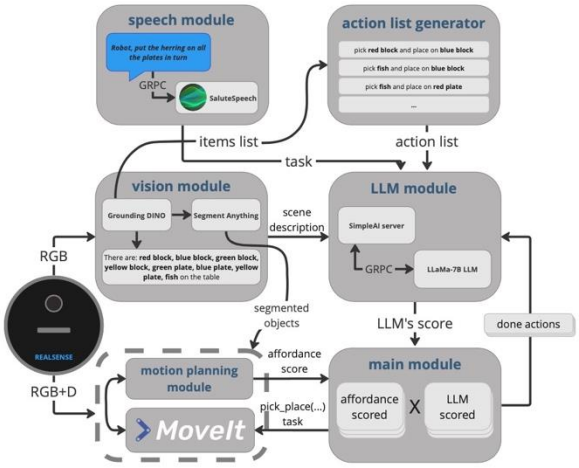


Рис. 3. Компонентная диаграмма робота-манипулятора с голосовым управлением [4]

Далее эта информация попадает в модуль получения навыка, который состоит из трех частей: использования LLM модели, выявление наиболее близкой операции к команде и выполнение этой команды. После выполнения команды результат возвращается в модель LLM, которая будет дальше разбивать команду.

B. *Архитектурные решения модулей LLM*

Остановимся более подробно на архитектуре модуля LLM (см. рис. 4).

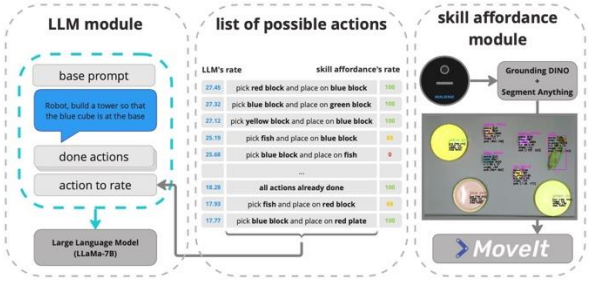


Рис. 4. Компонентная диаграмма модуля skill-affordance

Разработанный модуль получает в качестве входного информационного потока не только текстовую команду от пользователя, но и имеет базу знаний сцены, в которой работает робот-манипулятор. Хранимая база знаний

содержит перечень объектов сцены и их характеристик: цвет и габаритные размеры.

Особенность данной реализации заключается в том, что модуль LLM не используется для генерации токенов или для попытки каким-либо образом интерпретировать команду пользователя. С его помощью производится оценка реалистичности контекста выполнения команды в рамках существующей сцены робота.

На рис. 5 представлена архитектура робота, который реализует модуль LLM на основе модели GPT-2 от компании OpenAI [4]. Необходимо отметить, что в данном случае использовалась модель, с которой провели дополнительное обучение, что позволило с ее помощью реализовать напрямую генерацию двигательных функций робота.

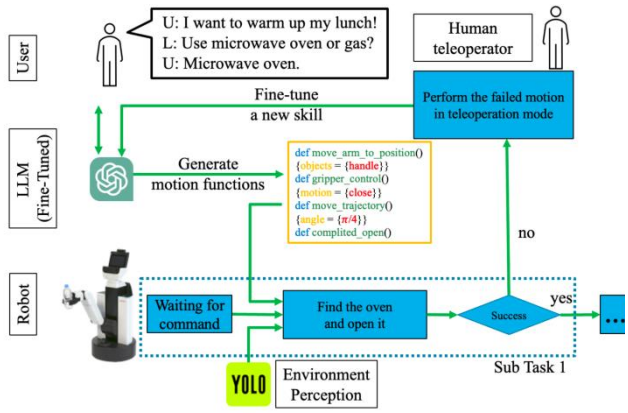


Рис. 5. Архитектура робота-манипулятора на основе GPT.

Компания Figure представила робота (см. рис. 6), в основе которого заложен схожий принцип [5], но с использованием мультимодальной LLM модели, которая способна принимать и обрабатывать информацию из различного рода источников: текст, голос, видео.

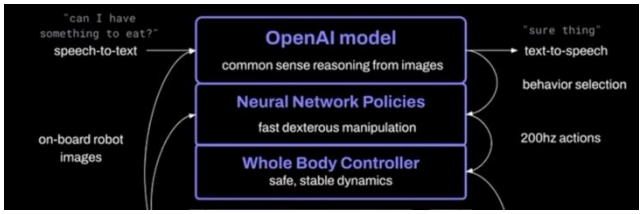


Рис. 6. Архитектура робота Figure на основе GPT

Все решения, полученные на основе моделей класса GPT, требуют лицензирования от компании OpenAI, в данной работе применено открытое программное обеспечение в виде модели LLaMA. Кроме того, модели класса LLaMA изначально создавались с расчетом на эффективность использования памяти и скорость взаимодействия в сценариях человек-робот наподобие чат-ботов. Поэтому модели класса LLaMA выигрывают в скорости относительно «тяжеловесных» моделей GPT общего назначения. Таким образом, использование модели LLaMA в ансамбле с дополнительными программно-аппаратными подсистемами робота не уступает использованию GPT в промышленных сценариях и имеет потенциал успешности внедрения в промышленную среду [7].

IV. Результаты

Для принятия решения об эффективности и целесообразности применения выбранной архитектуры управляющего модуля робота-манипулятора на основе модели LLaMA-7B произведена серия экспериментальных исследований на не промышленных сценах с оценкой, насколько языковая модель, обученная на массиве разнообразных данных, способна решать задачи в ограниченном пространстве ответов.

В результате для команды с точным лаконичным указанием управляющей зависимости и одной контекстной связи субъект-объект получен абсолютно правильный результат.

В сценариях, когда даны более одной контекстной зависимости субъект-объект робот может производить серию дополнительных действий, но конечный результат все равно верный. Например, желтый кубик уже лежит в синей тарелке и поступает голосовая команда: “Положи селедку в ту тарелку, где лежит желтый кубик” как показано на рис. 7 робот произвел два действия вместо одного. То есть, он сначала переложил кубик, а затем положил к нему рыбку.

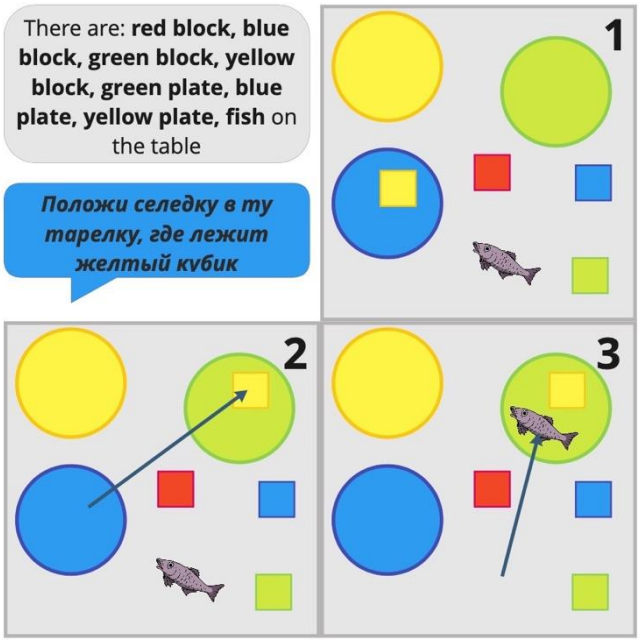


Рис. 7. Пример реальной задачи для робота от пользователя

В более контекстно свободных сценариях управляющих команд наблюдались некоторые неточности. Так, например, с командой “Собери башенку из кубиков так, чтобы красный блок оказался в основании” робот справляется верно, так же как с командами, содержащими описание последовательности действий: “Положи зеленый кубик на красный, а затем перенеси синий кубик на красную тарелку” или “Положи рыбку на все кубики по очереди”. А вот задачи на количество предметов, сложенных в высоту, вызывают трудности. Если, например, попросить построить башню высоты 2, модуль LLM выдаст два действия, а робот построит башню из трех кубиков.

V. Заключение

Разработанная архитектура робота-манипулятора показала в ходе экспериментов результаты с надежной точностью интерпретации управляющей команды в прямые действия робота над ограниченной сценой и объектами манипуляции. Таким образом, применимость моделей семейства LLaMA в задачах управления голосовыми командами робота-манипулятора показали свою состоятельность. Реализованный в рамках данной работы программно-аппаратный комплекс, как пример типового решения, является перспективным для современного рынка робототехники.

Литература

[1] Solodov, S.V., Mamai, I.B., Pronichkin, S.V., “Framing regional innovation and technology policies for transformative change”, IOP Conference Series: Earth and Environmental, 2022, 981(2), 022007, doi: 10.1088/1755-1315/981/2/022007

[2] Trofimov, V.B., Temkin, I.O., Solodov, S.V, “APPLICATION OF CASE-BASED REASONING IN HAZARD EVALUATION IN COMPLEX PROCESS FLOW CONTROL”, Eurasian Mining, 2023, 40(2), pp. 41–46, doi: 10.17580/em.2023.02.09

[3] Zoumana Keita. “Llama.cpp Tutorial: A Complete Guide to Efficient LLM Inference and Implementation”, Available at: https://www.datacamp.com/tutorial/llama-cpp-tutorial (Accessed: February 25, 2024)

[4] Haokun Liu, Yaonan Zhu, Kenji Kato, Izumi Kondo, Tadayoshi Aoyama, Yasuhisa Hasegawa. “LLM-Based Human-Robot Collaboration Framework for Manipulation Tasks”. Available at: https://arxiv.org/pdf/2308.14972.pdf (Accessed: September 17, 2023)

[5] OpenAI. “Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation”, Available at: https://community.openai.com/t/openai-chatgpt-robot-figure-01/681733 (Accessed: March 17, 2024)

[6] Arlazarov, V & Arlazarov, Vladimir & Bulatov, Konstantin & Chernov, Timofey & Nikolaev, Dmitry & Полевой, Дмитрий & Sheshkus, Alexander & Skoryukina, Natalya & Slavin, Oleg & Usilin, S. (2022). Mobile ID Document Recognition-Coarse-to-Fine Approach. Pattern Recognition and Image Analysis. 32. 89-108. 10.1134/S1054661822010023

[7] R. R. Bikmaev, M. D. Zolotov, A. N. Popov and R. N. Sadekov, "Improving the Accuracy of Supporting Mobile Objects with the Use of the Algorithm of Complex Processing of Signals with a Monocular Camera and LiDAR," 2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), St. Petersburg, Russia, 2019, pp. 1-4, doi: 10.23919/ICINS.2019.8769360.

[8] Savelyev, B.I., Solodov, S.V., Tropin, D.V., “Formalizing and securing relationships on multi-task metric learning for IoT-based smart cities”, Journal of Physics: Conference Series, 2021, 2094(3), 032062, doi: 10.1088/1742-6596/2094/3/032062

ГРНТИ 28.23.15
ASJC 1701
OECD 01.02.EP

Применение компьютерного зрения для распознавания автомобильных номеров

Д. В. Савенков
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m2312188@edu.misis.ru

Д. В. Лоткова
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m1908659@edu.misis.ru

***Аннотация* — в данной статье рассматривается применение методов компьютерного зрения для распознавания автомобильных номеров на изображениях и анализируются результаты их применения к реальным данным. Современные технологии машинного и глубокого обучения предоставляют инструменты для разработки эффективных алгоритмов, способных автоматически обнаруживать и распознавать номерные знаки на изображениях. Обзор существующих подходов к данной задаче включает в себя методы предобработки изображений, выбор архитектур нейронных сетей и использование современных моделей, таких как YOLOv8. В заключение рассматривается применимость этих методов на реальных данных и их потенциал в различных практических сценариях.**

Ключевые слова — компьютерное зрение, обнаружение объектов, распознавание автомобильных номеров, нейронные сети, YOLO, YOLOv8m.

I. Введение

Искусственные нейронные сети находят широкое применение в различных областях, таких как системы навигации [1], детекция образов [2], обработка естественного языка (ОЯЕ) [3], компьютерное зрение [4] и многие другие. Например, в задачах классификации изображений глубокие нейронные сети способны автоматически распознавать и классифицировать объекты на фотографиях. Искусственные нейронные сети также эффективно применяются в задачах прогнозирования, управления процессами и в робототехнике [5], что подчеркивает их универсальность и эффективность в различных областях применения.

В последние годы технологии компьютерного зрения достигли значительных успехов, предоставляя новые возможности для автоматизации и повышения эффективности различных процессов. Компьютерное зрение (Computer Vision, CV) — это область искусственного интеллекта, связанная с анализом изображений и видео. Она включает в себя набор методов, которые наделяют компьютер способностью «видеть» и извлекать информацию из увиденного.

Для решения разнообразных задач в области компьютерного зрения применяются методы машинного обучения, включая глубокое обучение, а также различные техники обработки изображений. Эти методы используются для выполнения таких задач, как распознавание объектов, классификация изображений, сегментация, восстановление трехмерных моделей, распознавание жестов и многие другие.

Одной из областей применения этих технологий является распознавание автомобильных номеров. Автоматическое распознавание номеров транспортных средств имеет множество применений, включая управление дорожным движением, системы контроля доступа, парковочные системы, а также в обеспечении безопасности и правопорядка.

Распознавание автомобильных номеров представляет собой сложную задачу, включающую несколько этапов: обнаружение транспортного средства, локализация номера на изображении, сегментация символов и их последующее распознавание [6]. Каждый из этих этапов требует применения передовых алгоритмов и методов глубокого обучения для обеспечения высокой точности и надежности.

Основной целью данной работы является исследование алгоритмов и методов распознавания автомобильных номеров на изображениях с применением современных технологий глубокого обучения. Практическая часть работы состоит в обучении нейронной сети на двух наборах данных — открытом датасете Drivers LPD и локальном наборе, созданном авторами. Это позволяет оценить применимость выбранной модели в условиях реальных данных и сравнить её точность и производительность на различных наборах изображений.

Современные методы компьютерного зрения, такие как свёрточные нейронные сети (CNN) и алгоритмы глубокого обучения, показали высокую эффективность в решении задач распознавания образов. Особое внимание привлекает нейронная сеть YOLO (You Only Look Once), которая позволяет выполнять обнаружение объектов в реальном времени с высокой точностью [7]. В особенности YOLOv8 выделяется среди передовых алгоритмов обнаружения объектов, который активно применяется для детекции объектов также в реальном времени. Благодаря высокой эффективности и широкому распространению, YOLOv8 становится предпочтительным выбором для точного определения и локализации объектов на изображениях.

II. НАБОРЫ ДАННЫХ

Для обучения и тестирования нейронной сети, рассматриваемой в данном исследовании, были использованы различные наборы данных, включая как локальные наборы, собранные авторами исследования, так и открытые наборы данных. Рассмотрим подробнее используемые открытые наборы.

A. Открытый набор данных. Drivers LPD

Набор данных Drivers LPD [8] содержит около 4583 кадров различных транспортных средств, снятых в разных странах при разном освещении и с разных ракурсов. Набор содержит номерные знаки из различных стран, но, в основном, фокусируются на европейские, русские, американские, ближневосточные, индийские автомобильные номера.

В наборе данных около 3000 изображений автомобилей с отмеченными номерными знаками, при условии, что на номере есть хотя бы один распознаваемый символ и он не виден через стекло другого автомобиля (или в отражении). Также есть около 1500 изображений, которые либо вообще не содержат автомобилей, либо содержат автомобили под таким углом или разрешением, что их номерные знаки не распознаются (рисунок 3). Это сделано для уменьшения ложных срабатываний.

Рис. 1. Примеры кадров из датасета Drivers LPD

Набор данных в свою очередь разбит на подвыборки для обучения, тестирования и валидации (таблица 1).

ТАБЛИЦА I. Подвыборки набора данных Drivers LPD

Тип выборки данных	**Количество изображений**
Train	3719 (81,15%)
Val	304 (6,63%)
Test	560 (12,22%)
all images	4583

B. Локальный набор данных.

Локальный набор данных [9] был создан специально для тестирования выбранного решения на реальных изображениях и последующего сравнения с результатами на открытом датасете. Данный набор включает 952 изображения, собранные из открытых источников, а также из личных архивов авторов. Все изображения были размечены вручную и подготовлены к работе в соответствии с требованиями задачи.

Особенность локального набора данных заключается в том, что он включает изображения с разнообразными условиями съёмки, что позволяет объективно оценить эффективность алгоритма в реальных сценариях. На рисунке 2 представлены примеры изображений из локального набора, демонстрирующие его содержание.

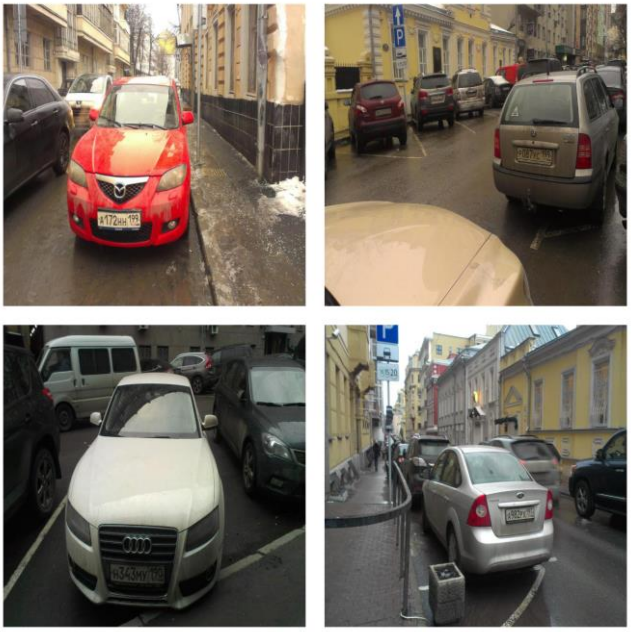


Рис. 2. Примеры кадров из локального датасета

Создание локального набора данных обеспечивает возможность проведения независимого тестирования и сравнения производительности модели на изображениях, которые лучше отражают условия её предполагаемого применения. Набор данных также в свою очередь разбит на подвыборки для обучения, тестирования и валидации (таблица 2).

ТАБЛИЦА II. Подвыборки локального набора данных

Тип выборки данных	**Количество изображений**
train	659 (70%)
val	189 (20%)
test	94 (10%)
all images	942

III. НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА

A. YOLO

Выделяют два основных метода детектирования объектов (рисунок 3) [10]:

- Метод, основанный на поиске регионов, содержащих объекты, и последующей классификации объектов внутри найденных

регионов. Задача детекции решается в два этапа (поиск и классификация), и такие методы называются «двухуровневые».

- Метод, основанный на решении задачи регрессии и классификации. Поиск областей и определение их классов происходит в один этап, а сам метод называется «одноуровневый».

К двухуровневым методам относится архитектура R-CNN (Regions With CNNs), которая состоит из трёх частей: CNN, регрессора ограничивающих рамок (bounding-box regressor) и классификатора на основе опорных векторов (SVM). Существуют и другие двухпроходные методы детекции с использованием нейронных сетей, которые работают быстрее и точнее благодаря улучшениям. Однако данные нейронные сети имеют два существенных недостатка: во-первых, они не анализируют изображение целиком, а рассматривают лишь отдельные регионы, во-вторых, их производительность относительно невысока.

К одноуровневым методам относятся следующие архитектуры нейронных сетей: SSD (Single Shot MultiBox Detector), RetinaNet, YOLO (You Only Look Once). Также внутри одной архитектуры существуют различные модификации моделей, например, разное количество свёрточных слоёв и количество параметров.

Архитектура YOLO обладает рядом преимуществ по сравнению с другими методами, избегая двух вышеупомянутых недостатков и демонстрируя высокую эффективность.

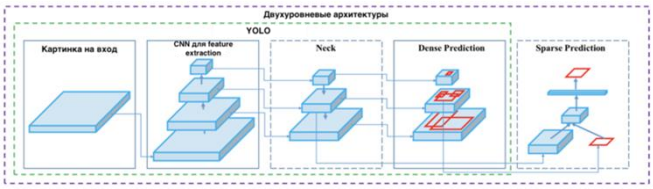


Рис. 3. Различие двухуровневых и одноуровневых архитектур

Архитектура YOLO на начальных этапах функционирования не сильно отличается по логике блоков от других детекторов. Входным сигналом является изображение, которое затем преобразуется в карты признаков (feature maps) с использованием свёрточной нейронной сети (CNN). В YOLO используется собственная CNN, известная как Darknet-53 (рисунок 4) [11]. Далее, эти карты признаков анализируются особым образом, что позволяет определить позиции и размеры ограничивающих рамок (bounding boxes) и классифицировать объекты, содержащиеся в этих рамках.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Рис. 4. Архитектура Darknet-53

YOLO (You Only Look Once) основывается на принципе однократного анализа изображения. За один проход изображения через нейронную сеть YOLO выполняет все необходимые операции по обнаружению объектов. Входной слой принимает изображение фиксированного размера, обычно 416x416 или 608x608 пикселей.

Основу архитектуры составляет свёрточная нейронная сеть (backbone), предназначенная для извлечения признаков. В различных версиях YOLO используются разные backbone сети, такие как Darknet-53 в YOLOv3. Для обработки признаков на разных уровнях и масштабах используется Feature Pyramid Network (FPN), что позволяет эффективно работать с объектами различных размеров.

На выходе сети находится головная часть, которая предсказывает bounding boxes, классы объектов и вероятности наличия объектов в каждой области. YOLO делит изображение на сетку, каждая ячейка которой отвечает за предсказание объектов, центр которых попадает в эту ячейку. Для предсказания bounding boxes используются якорные рамки (anchor boxes), причём каждая ячейка сетки предсказывает сдвиги относительно этих рамок.

Функция потерь в YOLO учитывает ошибки предсказания координат, размеров bounding boxes и классификации, оптимизируя сеть для улучшения детекции. Алгоритм постобработки Non-Maximum Suppression (NMS) устраняет избыточные предсказания и сохраняет рамки с наивысшей вероятностью [12].

Архитектура YOLO обеспечивает высокую скорость и точность детекции объектов. Различные версии YOLO включают улучшенные backbone сети, более точные функции потерь и усовершенствованные методы предсказания объектов.

B. YOLOv8m

Обнаружение объектов является более сложной задачей, чем классификация, которая может распознавать объекты, но не указывает их расположение на изображении и не работает с

изображениями, содержащими более одного объекта. YOLO позволяет одной CNN одновременно прогнозировать несколько ограничивающих рамок и классов для этих рамок. YOLO обучается на полных изображениях и напрямую оптимизирует производительность обнаружения.

YOLOv8 — это одна из последних итераций в серии детекторов объектов YOLO, предлагающая передовые характеристики в плане точности и скорости. Опираясь на достижения предыдущих версий YOLO. YOLOv8 предлагает новые функции и оптимизации, которые делают его идеальным выбором для решения различных задач по обнаружению объектов в широком спектре приложений.

Архитектура YOLOv8 представлена на рисунке 5. Сеть включает слои двумерной свёртки Conv2d, слои уменьшения размерности MaxPool2d, слои нормализации пакетов BatchNorm2d и слои активации SiLU, которые умножают входной сигнал на сигмоиду от входного сигнала. Эти слои сгруппированы по блокам, которые составляют основу сети [13].

Для Backbone представляет собой последовательность блоков Conv и C2f, выполняющих функции свёртки и нахождения карт признаков с пирамидой масштабов, которая используется для объединения признаков. Блоки Conv и C2f (coarse-to-fine) являются составными частями сети Backbone. SPPF (spatial pyramid pooling fast) — это пирамида масштабов, применяемая для объединения признаков на разных масштабах. Слой Upsample используется для повышения размерности карты признаков, что необходимо для согласования входных размеров.

Финальная последовательность слоёв, Head, формирует ограничивающие рамки и классы объектов с помощью блоков Detect. Блок Detect включает ряд слоёв с использованием свёртки с размером ядра 1, что позволяет уменьшать или увеличивать количество каналов. Подобные слои часто применяются в полностью свёрточных нейронных сетях на последних этапах сети.

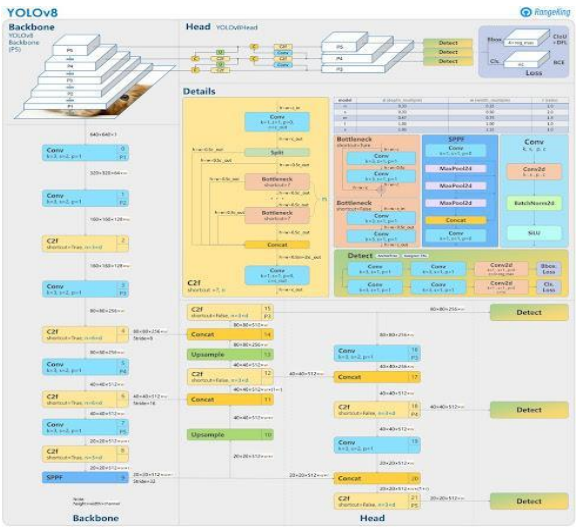


Рис. 5. Архитектура YOLOv8

Архитектурные улучшения в YOLOv8:

- Нейронная сеть CSPDarknet53. В YOLOv8 используется CSPDarknet53 в качестве основной сети. Это модифицированная версия Darknet-53, включающая блок "cross-stage" (CSP), что способствует повышению эффективности и обобщающей способности модели.
- Пирамидальная сеть (PANet). PANet используется для объединения признаков на разных уровнях, что улучшает обработку объектов различных масштабов.
- SPP (Spatial Pyramid Pooling). В YOLOv8 применяется SPP для увеличения размера поля зрения и улучшения обнаружения мелких объектов [14].
- Безъякорная сплит-головка Ultralytics. YOLOv8 использует безъякорную сплит-головку Ultralytics, что способствует повышению точности и эффективности процесса обнаружения по сравнению с подходами, основанными на якорях.
- YOLOv8 представлена в различных конфигурациях: YOLOv8-S, YOLOv8-M, YOLOv8-L и YOLOv8-XL. Каждая конфигурация отличается архитектурными характеристиками, такими как количество слоёв и параметров, что позволяет выбирать модель в зависимости от требований к производительности и точности.

В каждой категории моделей YOLOv8 есть пять моделей для обнаружения, сегментации и классификации. YOLOv8 Nano - самый быстрый и маленький, в то время как YOLOv8 Extra Large (YOLOv8x) - самый точный, но самый медленный среди них.

Приведённый ниже график (рисунок 7), демонстрирующий карту трёх моделей при 0,50 IoU,дает более ясное представление [15].

Для оценки обнаружения объектов применяется метрика IoU (Intersection over Union), которая сравнивает две ограничивающие рамки путем вычисления отношения площади их пересечения к площади их объединения (рисунок 6). Метрика IoU лежит в диапазоне [0, 1] и чем больше ее значение, тем сильнее совпадают ограничивающие рамки.

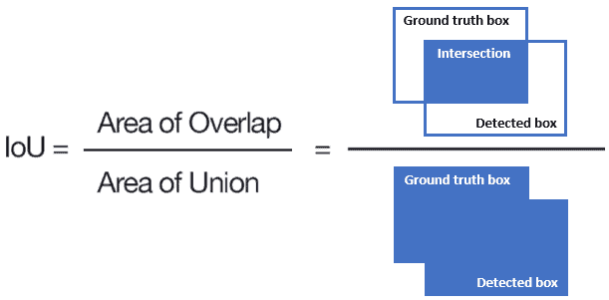


Рис. 6. Метрика IoU (Intersection over Union)

За исключением модели YOLOv8 Nano, остальные две модели показывают постоянное улучшение в ходе

обучения. Продолжение обучения этих двух моделей может привести к ещё более высоким результатам.

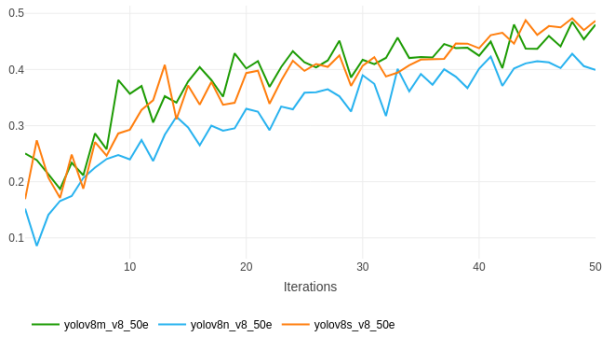


Рис. 7. Сравнение между картой модели YOLOv8 Nano, Small и Medium при 0,50 IoU

YOLOv8m представляет собой компромисс между лёгкими моделями (YOLOv8n) и более тяжёлыми (YOLOv8s), обеспечивая лучшее соотношение скорости и точности. Она достаточно быстра, но при этом предлагает более высокую точность по сравнению с YOLOv8n. Благодаря своей архитектуре YOLOv8m способна лучше справляться с детекцией объектов в сложных сценах, где присутствует множество объектов различных размеров и типов.

IV. Оценка точности

Для оценки эффективности модели мы использовали несколько метрик. Для анализа производительности модели были получены (рисунок 9):

- Train Box Loss (TBL) отражает различие между предсказанными ограничивающими рамками и реальными рамками объектов в учебных данных. Меньшая потеря означает более точное соответствие предсказанных моделью рамок реальным рамкам.
- Train Class Loss (TCL) оценивает разницу между предсказанными вероятностями классов и реальными метками классов объектов в учебных данных. Меньшая потеря класса указывает на более точное соответствие предсказанных моделью вероятностей классов реальным меткам классов.
- Train DFL Loss (TDFL) измеряет различие между предсказанными картами признаков и реальными картами признаков объектов в учебных данных. Меньшая потеря DFL указывает на более точное соответствие предсказанных моделью карт признаков реальным картам признаков.
- Метрика Precision (1) измеряет долю правильно обнаруженных областей среди всех предсказанных ограничивающих рамок. Более высокая точность указывает на лучшую идентификацию верно обнаруженных областей и минимизацию ложноположительных результатов.

$$Precision = \frac{TP}{T+FP} \quad (1)$$

- Метрика Recall (2) измеряет долю правильно обнаруженных областей среди всех фактических ограничивающих рамок. Более высокая полнота указывает на лучшую идентификацию всех правильно обнаруженных областей и минимизацию ложноотрицательных результатов.

$$Recall = \frac{TP}{T+FN} \quad (2)$$

- Метрика mAP50 (B) измеряет среднюю общую точность модели по разным категориям объектов при пороге пересечения-объединения (IoU) 50%. Более высокое значение mAP50 указывает на лучшую идентификацию и локализацию объектов разных категорий.
- Метрика mAP50-95 (B) измеряет среднюю общую точность модели по разным категориям объектов при порогах IoU от 50% до 95%. Более высокое значение mAP50-95 указывает на лучшую идентификацию и локализацию объектов различных категорий при широком диапазоне порогов IoU.

Результаты при обучении модели на наборе данных Drivers LPD представлены на рисунке 8, также графики были получены и проанализированы для локального датасета (рисунок 9).

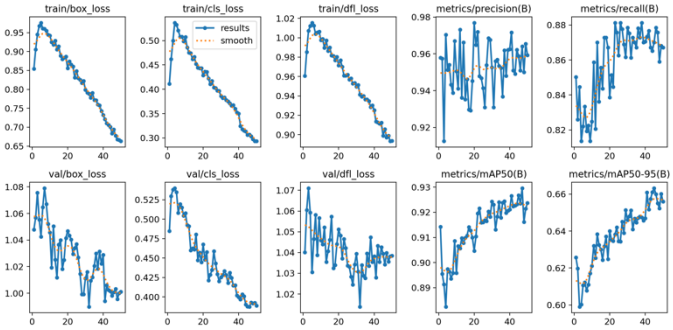


Рис. 8. Результаты при обучении модели. Drivers LPD

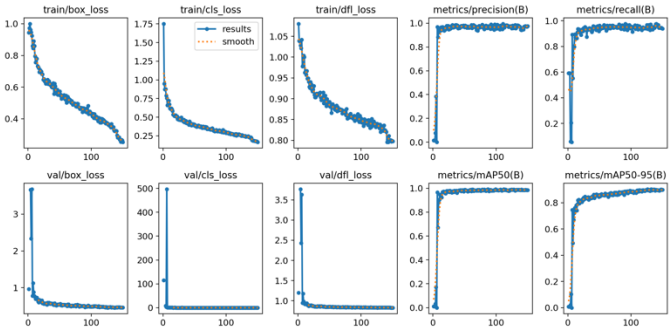


Рис. 9. Результаты при обучении модели. Локальный датасет

Метрика F1 Score (3) представляет собой гармоническое среднее между точностью и полнотой. Она учитывает показатели полноты и точности, что позволяет избежать переоценки модели, которая может демонстрировать высокие результаты по одному из этих критериев, но низкие - по другому.

$$F1 = 2 * Precision * \frac{Recall}{Precision+Recall} \quad (3)$$

Обучение нейронной сети на датасетах Drivers LPD и локальном наборе (рисунок 10) данных выявило, что, несмотря на значительно худшие результаты потерь (TBL, TCL, TDFL) на открытом наборе данных, его F1-мера оказалась выше (0.95 против 0.92). Второй датасет продемонстрировал более высокую точность в отдельных компонентах задачи :в определении границ объектов, классификации и извлечении признаков (меньшие значения TBL, TCL и TDFL).

В таблице III представлена информация о полученных данных в ходе практической части.

ТАБЛИЦА III. Оценка детектирующей части для разных наборов данных

	Drivers LPD	**Локальный набор данных**
TBL	0.79	0.51
TCL	0.42	0.37
TDFL	0.95	0.87
Precision	0.97	0.93
Recall	0.93	0.91
F1	0.95	0.92

Однако более высокая F1-мера для Drivers LPD свидетельствует о лучшем балансе между точностью и полнотой предсказаний. Это несоответствие может быть связано с особенностями данных: открытый набор данных (рисунок 11), вероятно, характеризуется более сбалансированным составом классов или лучшей репрезентативностью объектов, что способствовало более эффективной генерализации модели.

Рис. 10. Примеры полученных изображений из локального набора данных

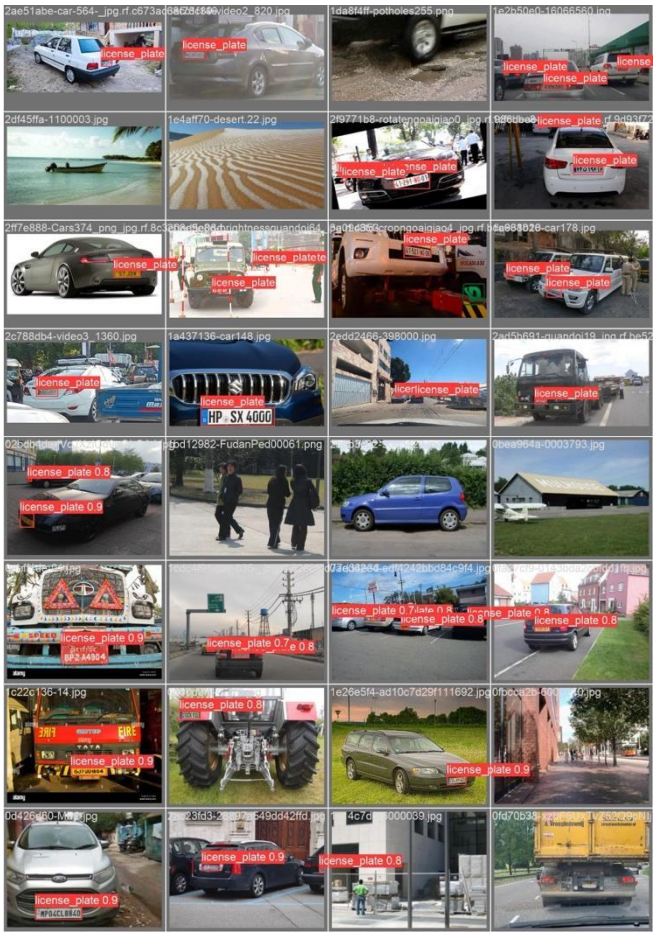


Рис. 11. Примеры полученных изображений из датасета Drivers LPD

Исходя из F1 метрики на первом датасете, можно заметить, что данный набор данных показывает себя лучше, чем локальный на 0.03. На тестовой выборке номерные рамки угадывались лучше, потому что количество фотографий было больше при обучении модели при помощи первого датасета. Для этого датасета модели показали хорошие результаты, которые удовлетворяют решению задачи.

На основании полученных данных моделей можно сделать вывод, что датасет Drivers LPD показывает лучшие результаты, чем локальный набор данных.

V. Заключение

Были рассмотрены основные наборы данных, на которых обучалась и тестировалась рассматриваемая нейронная сеть. Исследованы два подхода к детектированию объектов, а также изучена архитектура YOLO и изменения, внесенные в следующую версию YOLOv8. Были рассмотрены различные категории моделей, каждая из которых проанализирована с точки зрения её архитектуры, процесса обучения, а также используемых наборов данных для обучения и тестирования.

Основное отличие YOLO от других алгоритмов свёрточных нейронных сетей (CNN), используемых для

обнаружения объектов, заключается в его высокой скорости распознавания объектов. YOLO обрабатывает изображение целиком, проходя его через свёрточную нейронную сеть лишь один раз, что и объясняет название "You Only Look Once". В других алгоритмах этот процесс повторяется многократно, что значительно замедляет их работу. Благодаря этому YOLO обладает значительным преимуществом в скорости обнаружения объектов по сравнению с другими алгоритмами.

Модель YOLOv8m, входящая в восьмую версию этой серии, обеспечивает баланс между производительностью и вычислительными ресурсами, что делает её особенно подходящей для практического применения в задачах распознавания автомобильных номеров.

Модель YOLOv8m была протестирована на датасетах Drivers LPD и локальном наборе данных. Отдельно были оценены качество определения и классификации автомобильных номеров. По полученным данным очевидно, что нейронная сеть YOLOv8m, отлично справляется с поставленной задачей, что объясняется разными обучающими процессами и разным качеством обучающих выборок.

Литература

[1] D. B. Pazychev and R. N. Sadekov, "Simulation of INS Errors of Various Accuracy Classes," 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), St. Petersburg, Russia, 2020, pp. 1-3

[2] Баканов П.П., Измайлов Л.С., Тригуб Н.А. ФОРМИРОВАНИЕ ЧИСЛОВОГО КОДА ФРАКТАЛЬНОЙ СТРУКТУРЫ ТЕКСТУРИРОВАННОГО ОПТИЧЕСКИ АНИЗОТРОПНОГО ГЛАСТЭЛИТА // Перспективы науки . - 2023. - №5. - С. 118-125.

[3] Berdichevskaia A. Atypical lexical abbreviations identification in Russian medical texts //2022 12th International Conference on Pattern Recognition Systems (ICPRS). – IEEE, 2022. – С. 1-5.

[4] R. R. Bikmaev, M. D. Zolotov, A. N. Popov and R. N. Sadekov, "Improving the Accuracy of Supporting Mobile Objects with the Use of the Algorithm of Complex Processing of Signals with a Monocular Camera and LiDAR," 2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), St. Petersburg, Russia, 2019, pp. 1-4, doi: 10.23919/ICINS.2019.8769360.

[5] Практическое применение роботов и сопутствующих технологий в борьбе с пандемией COVID-19 / А. Р. Ефимов, А. С. Гонноченко, Д. Б. Пайсон [и др.] // Робототехника и техническая кибернетика. – 2020. – Т. 8, № 2. – С. 87-100.

[6] Болотова Юлия Александровна, Спицын Владимир Григорьевич, Рудометкина Моника Николаевна Распознавание автомобильных номеров на основе метода связных компонент и иерархической временной сети // КО. 2015. №2. URL: https://cyberleninka.ru/article/n/raspoznavanie-avtomobilnyh-nomerov-na-osnove-metoda-svyaznyh-komponent-i-ierarhicheskoy-vremennoy-seti (Accessed: May 15, 2024).

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. “YouOnly Look Once: Unified, Real-Time Object Detection” (9 May 2016)

[8] Drivers LPD, available at: https://www.kaggle.com/datasets/fxmikf/diverse-lpd-training-ready (Accessed: May 10, 2024).

[9] Подготовленный локальный набор данных. RuNumbers, available at:https://universe.roboflow.com/dzane/runumbers/model/1 (Accessed: May 10, 2024).

[10] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, Xindong Wu, “ObjectDetection With Deep Learning: A Review”, IEEE Transactions onNeural Networks and Learning Systems, vol. 99, pp 1-22, 2019.

[11] “A Comprehensive Review of YOLO: From YOLOv1 and Beyond” available at https://arxiv.org/pdf/2304.00501.pdf (Accessed: May 03, 2024).

[12] “YOLOv3: An Incremental Improvement” available at: arXiv:1804.02767 (Accessed: May 12, 2024).

[13] “Ultralytics YOLOv8” available athttps://github.com/ultralytics/ultralytics (Accessed December 19,2023)

[14] Jianbiao Mei, Yu Yang, Mengmeng Wang, Xiaojun Hou, Laijian Liand Yong Liu. “PANet: LiDAR Panoptic Segmentation with SparseInstance Proposal and Aggregation”

[15] “Распознавание номерных знаков. Как все ускорить”, available at: https://habr.com/ru/articles/594401/ (Accessed: May 10, 2024).

[16] Richard Evans. “Confusion Matrices and Accuracy Statistics forBinary Classifiers Using Unlabeled Data: The Diagnostic TestApproach”

ГРНТИ 20.53.19
ASJC 1707
OECD 01.02.EP

Исследование возможности детектирования и классификации видов транспорта

И. Д. Фомин
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2415488@edu.misis.ru

М. А. Омеров
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2009187@edu.misis.ru

Аннотация — в данной работе проводится сравнительный анализ моделей YOLOv5 и YOLOv8 в задачах детекции различных видов транспортных средств, включая автомобили, автобусы, грузовики, мотоциклы, рикши и минивэны. Сравнение осуществляется на основе экспериментального обучения и тестирования на специализированных наборах данных, с учетом метрик точности (mAP), скорости обработки и вычислительных ресурсов. Результаты анализа подчеркивают преимущества и недостатки каждой из моделей в различных условиях применения.

Ключевые слова — компьютерное зрение, YOLOv5, YOLOv8, детекция транспорта, автономные транспортные средства, mAP, набор данных

I. ВВЕДЕНИЕ

С развитием технологий искусственного интеллекта и компьютерного зрения особое внимание уделяется разработке систем, способных эффективно и точно решать задачи, связанные с автоматизацией управления дорожным движением и созданием беспилотных транспортных средств. Эти системы применяют алгоритмы детекции и распознавания объектов для анализа дорожной обстановки в реальном времени, что позволяет автономным автомобилям безопасно передвигаться в сложных условиях городского и шоссейного движения. Кроме того, аналогичные алгоритмы активно используются в беспилотных летательных аппаратах (БПЛА) [1] для мониторинга транспортной инфраструктуры, наблюдения за дорожной обстановкой с воздуха и выполнения спасательных операций в зонах стихийных бедствий. С середины 1980-х годов ведущие университеты, научно-исследовательские центры и промышленные компании активно разрабатывают решения для беспилотных автомобилей и БПЛА, включая такие компании, как Tesla, Mercedes, Honda, а также IT-гиганты Google, Яндекс и Uber.

Детекция и распознавание объектов дорожной сцены являются ключевыми задачами при создании систем управления автономных транспортных средств [2]. Эти задачи включают в себя идентификацию автомобилей, автобусов, мотоциклов, велосипедов и других объектов, взаимодействующих в транспортной среде [3]. Технологии глубокого обучения и компьютерного зрения обеспечивают высокую точность и производительность при решении подобных задач, демонстрируя возможность работы в реальном времени и адаптацию к различным условиям освещения и погодным факторам [4].

Особый интерес представляют алгоритмы семейства YOLO (You Only Look Once), которые получили широкое распространение благодаря их скорости и точности [5]. Различные версии YOLO, такие как YOLOv5 и YOLOv8, представляют собой мощные инструменты для решения задач обнаружения и классификации объектов [6]. Несмотря на то, что обе модели зарекомендовали себя как высокоэффективные решения, их сравнительные характеристики в специфических условиях задач, связанных с детекцией транспорта, остаются актуальной темой для исследования.

В данной работе проводится сравнительный анализ моделей YOLOv5 и YOLOv8 в задачах детекции различных видов транспортных средств. Исследование направлено на выявление сильных и слабых сторон каждой модели, а также оценку их производительности на основе ключевых метрик, таких как точность, полнота и средняя точность (mAP) [7, 8]. Для экспериментов используется специализированный набор данных с аннотациями транспортных средств, что позволяет провести объективное сравнение моделей в условиях, приближенных к реальной эксплуатации.

II. НАБОРЫ ДАННЫХ

Для обучения и тестирования моделей YOLOv5 и YOLOv8, рассматриваемых в данной работе, использовался один и тот же открытый набор данных [9]. Рассмотрим его подробнее.

Датасет содержит 3000 изображений транспортных средств, равномерно распределенных по 6 классам: Car, Threewheel, Bus, Truck, Motorbike, Van. Каждому классу соответствует по 500 изображений, что обеспечивает сбалансированность данных. Датасет представлен в формате YOLO (txt), что упрощает процесс интеграции с рассматриваемыми моделями.

Для обучения и валидации данных было применено стандартное разбиение: 70% (2100 изображений) — обучающая выборка и 30% (900 изображений) — валидационная.

На рисунке 1 представлены примеры изображений из набора данных для каждого класса,

а

демонстрирующие разнообразие условий съемки, включая различное освещение и точки обзора. Этот набор данных позволяет моделям YOLO эффективно изучать особенности транспортных средств, что делает его подходящим для задач детекции в реальных условиях эксплуатации.

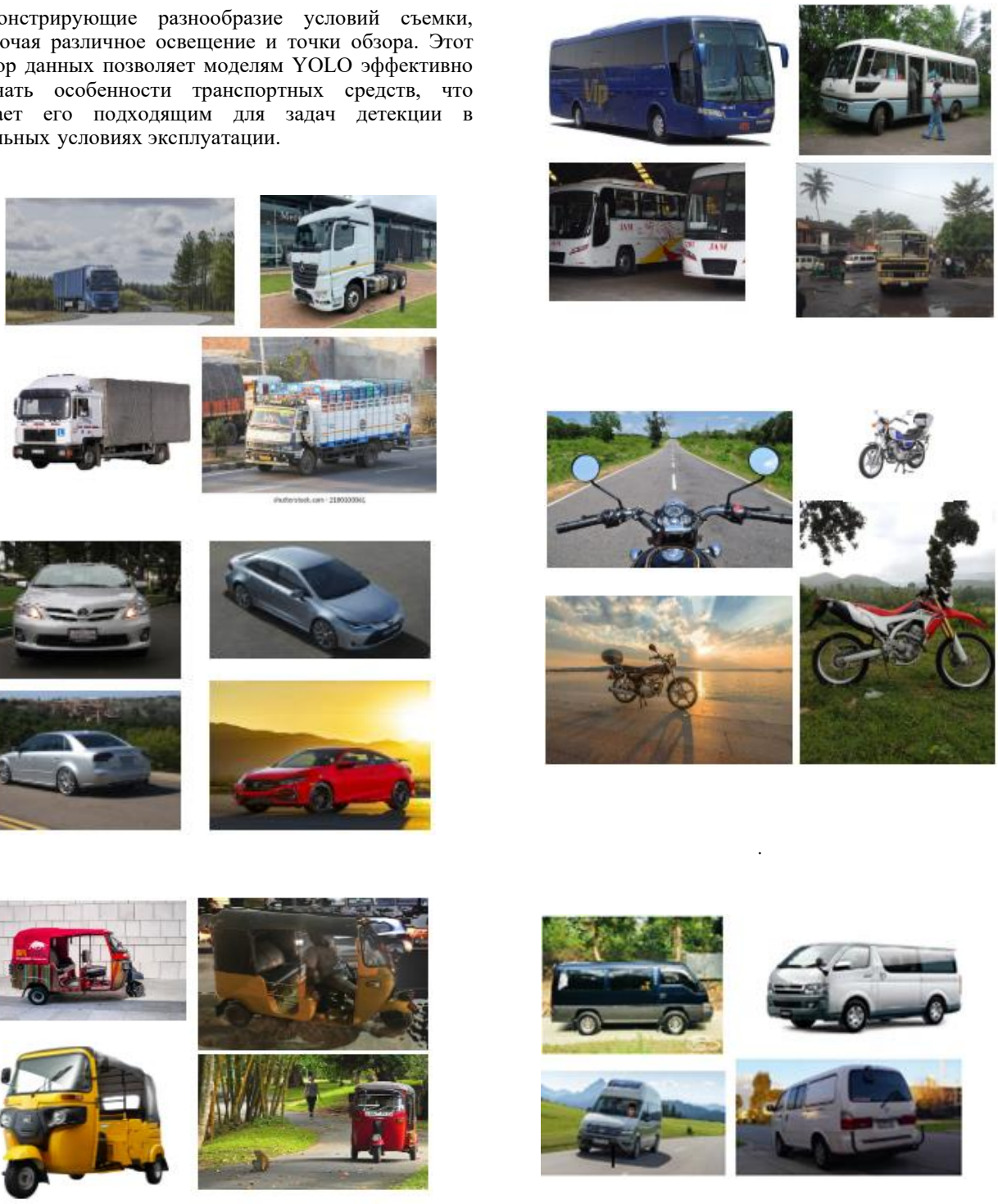


Рис. 1. Примеры изображений для каждого класса:
а) Car, б) Threewheel, в) Bus, г) Truck, д) Motorbike, е) Van

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

A. YOLOv5

YOLOv5, выпущенная в 2020 году компанией Ultralytics, стала одним из наиболее широко используемых инструментов для задач детекции объектов, благодаря своей простоте и производительности. Она предложила модульную архитектуру с использованием CSPNet (Cross Stage Partial Network) в качестве основы, что улучшило передачу признаков и уменьшило вычислительные затраты. YOLOv5 поддерживает несколько версий моделей — от компактной YOLOv5n (nano) до мощной YOLOv5x (extra-large) — обеспечивая гибкость для приложений с разными требованиями к ресурсам. Кроме того, YOLOv5 изначально поддерживала только задачи детекции объектов, но благодаря легкости настройки и активной поддержке со стороны разработчиков, быстро завоевала популярность в сообществе машинного обучения. Архитектура YOLOv5 представлена на рисунке 2 [10].

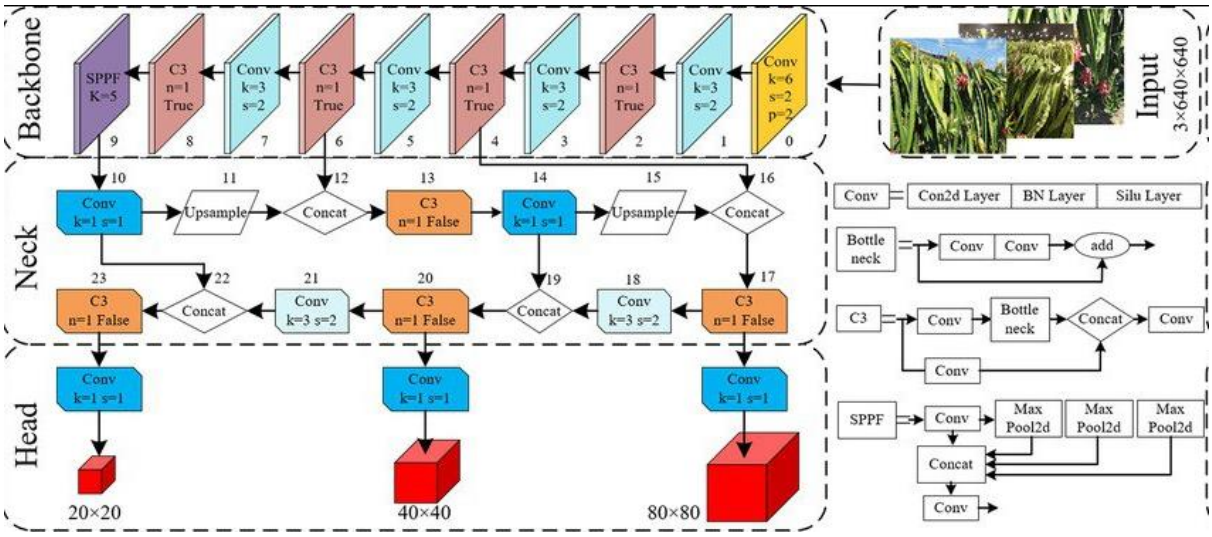


Рисунок 2. Архитектура YOLOv5

B. YOLOv8

YOLOv8 представляет собой улучшенную архитектуру свёрточных нейронных сетей, предназначенную для повышения точности и производительности в задачах компьютерного зрения. Модель поддерживает обнаружение объектов, сегментацию, оценку позы, отслеживание и классификацию. В YOLOv8 предложены пять версий, различающихся по масштабам и вычислительным требованиям: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) и YOLOv8x (extra-large) [11].

На рисунке 3 показана подробная архитектура YOLOv8 [12]. YOLOv8 использует ту же основу, что и YOLOv5, с некоторыми изменениями на CSP-слое, который теперь называется модулем C2f. Модуль C2f (кросс-стадийное частичное узкое место с двумя свертками) объединяет высокоуровневые признаки с контекстной информацией для повышения точности обнаружения.

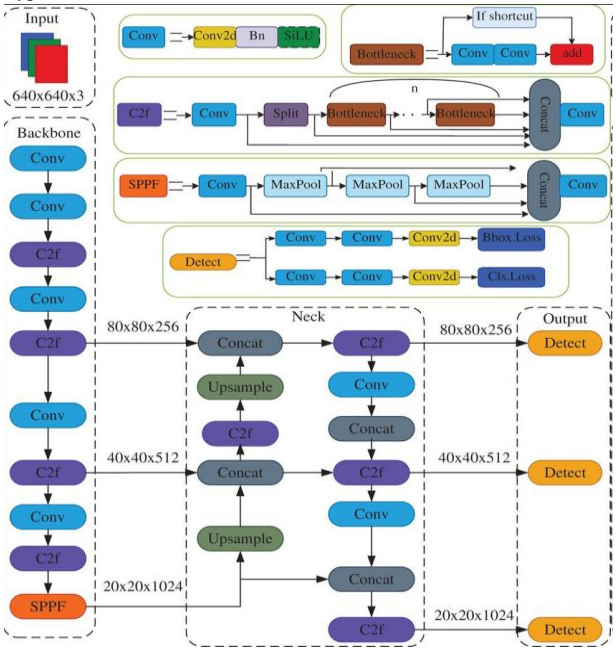


Рисунок 3. Архитектура YOLOv8

IV. СРАВНЕНИЕ

Было проведено обучение моделей YOLOv5 и YOLOv8. Для обучения использовался локальный набор данных, который был разбит на тренировочную и валидационную выборки в соотношении 7:3. Качество работы модели оценивалось как для локализации объекта, так и для

его классификации. Использовались следующие меры:

- TP – детектор верно локализовал транспортное средство и определил его класс.
- FP – детектор нашёл транспортное средство там, где его нет, или неверно определил его класс.
- FN – детектор не нашёл транспортного средства, хотя оно есть и для него есть разметка.

По введенным величинам строятся такие функции оценок, как:

- Точность – сколько раз модель обнаружила транспортное средство там, где оно действительно есть, к общему числу детектированных транспортных средств:

$$Precision = \frac{TP}{TP + FP}$$

- Полнота – сколько транспортных средств обнаружила модель от общего числа транспортных средств:

$$Recall = \frac{TP}{TP + FN}$$

- F1-мера – гармоническое среднее между точностью и полнотой, если один из параметров стремится к нулю, она также стремится к нулю:

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Mean Average Precision (mAP) [13] - популярная метрика оценки в задаче обнаружения объектов, включая модель YOLO. Она используется для оценки точности модели обнаружения объектов путем измерения ее способности обнаруживать объекты на изображении, а также точности обнаружения. mAP учитывает как количество правильно обнаруженных объектов, так и качество обнаружения, что делает ее надежной метрикой для оценки производительности моделей обнаружения объектов.

В YOLO mAP особенно важна, так как она измеряет точность модели в обнаружении интересующих объектов. Чем выше значение mAP, тем лучше модель способна идентифицировать объекты на изображении. Поскольку YOLO является моделью обнаружения объектов, разработанной для реального времени, достижение высоких значений mAP критически важно, чтобы модель точно обнаруживала объекты в реальных сценариях. Высокое значение mAP указывает на то, что модель может эффективно идентифицировать объекты и может быть использована с уверенностью в реальных приложениях.

Матрица ошибок [14] является важным инструментом для оценки точности алгоритмов детектирования объектов, таких как YOLO. Матрица ошибок представляет собой таблицу, которая обобщает верные положительные, верные отрицательные, ложные положительные и ложные отрицательные предсказания, сделанные моделью, что помогает глубже понять ее работу. В рамках данного проекта используются две матрицы ошибок, представленные на рисунке 4 и рисунке 5 .

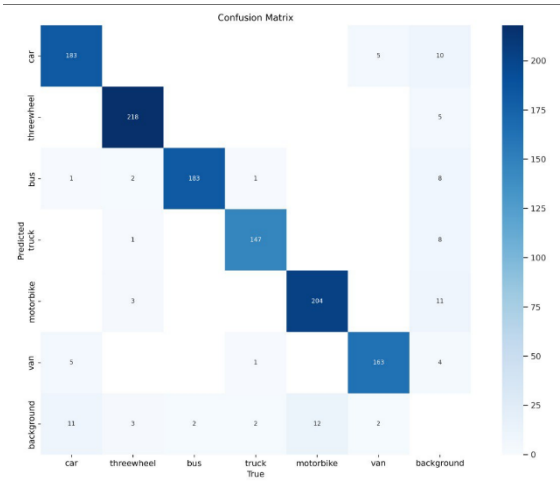


Рисунок 4. Матрица ошибок YOLOv5

Рисунок 5. Матрица ошибок YOLOv8

ТАБЛИЦА I. Сравнение детектирующей части.

	YOLOv5	YOLOv8
Precision	0,95	0,98
Recall	0,90	0,96
F1	0,92	0,97
mAP50	0,95	0,98

	YOLOv5	YOLOv8
Precision	0,95	0,98
Recall	0,90	0,96
mAP50-95	0,84	0,92

На основе данной таблицы можно сделать следующие выводы о сравнении моделей YOLOv5 и YOLOv8 по основным метрикам:

- Точность (Precision): YOLOv8 демонстрирует более высокую точность (0.98) по сравнению с YOLOv5 (0.95). Это указывает на меньшую вероятность ложных срабатываний у YOLOv8, что делает ее более надежной при детекции объектов.
- Полнота (Recall): YOLOv8 также превосходит YOLOv5 по полноте: 0.96 против 0.90. Это свидетельствует о способности YOLOv8 находить большее количество объектов из общего числа присутствующего в изображениях.
- F1-мера: значение F1-меры, объединяющей Precision и Recall, у YOLOv8 равно 0.97, что выше, чем у YOLOv5 (0.92). Это подтверждает общее превосходство YOLOv8 в сбалансированности между точностью и полнотой.
- Средняя точность (mAP50): при пороге IoU 50% YOLOv8 показывает результат 0.98, в то время как YOLOv5 — 0.95. Это говорит о том, что YOLOv8 точнее определяет местоположение объектов при низком пороге совпадения.
- Средняя точность (mAP50-95): для диапазона IoU от 50% до 95% YOLOv8 также превосходит YOLOv5: 0.92 против 0.84. Это указывает на лучшее обобщение YOLOv8 и ее способность точнее находить объекты в условиях более строгих порогов.

V. ЗАКЛЮЧЕНИЕ

В данной работе было проведено исследование и сравнительное тестирование двух современных моделей глубокого обучения — YOLOv5 и YOLOv8 — для задачи детекции различных видов транспорта. Анализ показал, что YOLOv8 значительно превосходит YOLOv5 по всем основным метрикам, включая Precision, Recall, F1-мера, mAP50 и mAP50-95. Достигнутое улучшение связано с более оптимизированной архитектурой YOLOv8, использованием современных подходов к предобработке данных и повышенной эффективностью обучения.

В ходе экспериментов использовался набор данных, включающий изображения различных типов транспорта, подготовленных в формате YOLO. Результаты показали, что YOLOv8 обеспечивает более высокую точность и полноту распознавания, что делает её более подходящей для задач компьютерного зрения в реальных приложениях, таких как системы беспилотных транспортных средств.

Несмотря на достигнутые результаты, стоит отметить, что производительность моделей может варьироваться в зависимости от сложности и особенностей данных, а также от аппаратных ресурсов. Будущие исследования могут быть направлены на адаптацию моделей к более сложным сценариям, например, детекцию транспорта в условиях слабого освещения, плохой видимости или высокой плотности объектов.

Таким образом, результаты работы демонстрируют значительный прогресс в области детекции объектов с использованием глубокого обучения и подчеркивают перспективность внедрения YOLOv8 в прикладные задачи компьютерного зрения.

ЛИТЕРАТУРА

[1] B. Ali, R. N. Sadekov, V. V. Tsodokova, “A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems,” Gyroscopy and Navigation, vol. 30, pp. 87–105, 10.17285/0869-7035.00105.

[2] N. S. Guzhva, V. E. Prun, V. V. Postnikov, M. G. Lobanov, R. N.Sadekov and D. L. Sholomov, "Using 3D Object Detection DNN in an Autonomous Tram to Predict the Behaviour of Vehicles in the Road Scene," 2022 29th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388.

[3] He, K., et al. "Deep Residual Learning for Image Recognition." — CVPR, 2016.

[4] Goodfellow, I., Bengio, Y., Courville, A. "Deep Learning." — MIT Press, 2016.

[5] Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M. "YOLOv4: Optimal Speed and Accuracy of Object Detection." — 2020.

[6] Jocher, G., et al. "YOLOv5 Documentation." — Ultralytics, 2021.

[7] Wang, C.-Y., et al. "YOLOv7: You Only Look Once at Accuracy and Speed." — 2022.

[8] Nadin Pethiyagoda. Vehicle Dataset for YOLO [Электронный ресурс]. – URL: https://www.kaggle.com/datasets/nadinpethiyagoda/vehicle-dataset-for-yolo (дата обращения: 28.12.2024).

[9] Ultralytics. YOLO Performance Metrics [Электронный ресурс]. – URL: https://docs.ultralytics.com/ru/guides/yolo-performance-metrics/#visual-outputs (дата обращения: 28.12.2024).

[10] Network structure diagram of YOLO V5 model [Электронный ресурс]. – URL: https://www.researchgate.net/figure/Network-structure-diagram-of-YOLO-V5-model_fig3_364596524?__cf_chl_tk=6Xb7DYgGj4NwHGEy18P33eXgfI9gwQehEb7kcTh5e0Y-1735448418-1.0.1.1-DT5Sz2.7d6ZoMJWElfcFm9G2DwToD06DEQIJptVxmuc (дата обращения: 28.12.2024).

[11] Исследование возможности распознавания объектов на спутниковых снимках. Available from: https://www.researchgate.net/publication/376809371_Issledovanie_vozmoznosti_raspoznavania_obektov_na_sputnikovyh_snimkah

[12] Model structure of the YOLO v8 algorithm [Электронный ресурс]. – URL: https://www.researchgate.net/figure/Model-

structure-of-the-YOLO-v8-algorithm_fig1_383187669 (дата обращения: 28.12.2024).
[13] Paul Henderson, Vittorio Ferrari. “End-to-end training of object class detectors for mean average precision” (12 Jul 2016)
[14] Richard Evans. “Confusion Matrices and Accuracy Statistics for Binary Classifiers Using Unlabeled Data: The Diagnostic Test Approach” (26 Aug 2022)

ГРНТИ 28.23.15; 67.25.19
ASJC 1707
OECD 1.02 EP

Исследование возможности детектирования курьеров доставки еды

М. А. Хижняк
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2414908@edu.misis.ru

***Аннотация*— в работе предлагается нейронная сеть для детектирования курьеров с изображений камер. Предложенное решение призвано помочь ресторанам с формированием общей оценки спроса, а также градостроителям и правовым регуляторам по созданию законов и рекомендаций по улучшению городской инфраструктуры. Основное внимание уделено разработке и обучению нейронной сети для распознавания курьеров по доставке еды, включая анализ их внешнего вида, униформы и других атрибутов (например, брендированных рюкзаков). Для этого была собрана и аннотирована специализированная база данных изображений, содержащих курьеров различных сервисов доставки в различных условиях: на улице, в транспорте и внутри помещений. Процесс включал этапы предварительной обработки данных, аугментацию, настройку гиперпараметров модели YOLO и её последующее тестирование. Проведены эксперименты, подтверждающие хорошую точность и высокую скорость детектирования, что делает предложенный подход применимым в задачах мониторинга, аналитики и автоматизации процессов в городской среде.**

Ключевые слова — компьютерное зрение, детекция, распознавание людей, распознавание курьеров, беспилотные автомобили, YOLO

I. Введение

Современные города становятся всё более сложными и динамичными системами, что требует применения новых подходов к их управлению и развитию. Искусственный интеллект (ИИ) активно используется в градостроительстве и анализе данных для решения задач, связанных с планированием инфраструктуры, оптимизацией транспортных потоков, мониторингом окружающей среды и повышением качества жизни горожан [1][2]. Технологии компьютерного зрения и анализа данных позволяют собирать, обрабатывать и интерпретировать большие объёмы информации в реальном времени, что открывает новые возможности для управления городскими процессами [3].

Курьерская деятельность, особенно в сфере доставки еды, является одной из наиболее быстрорастущих отраслей в России, в частности в Москве. Увеличение числа онлайн-заказов и удобство доставки сделали эту услугу неотъемлемой частью повседневной жизни горожан. Высокая концентрация курьеров в мегаполисах требует разработки новых подходов к мониторингу их деятельности и анализу данных. Это необходимо как для удовлетворения потребностей бизнеса, так и для улучшения городской инфраструктуры, снижения транспортной нагрузки и обеспечения безопасности на дорогах [4][5].

Основная задача данного исследования заключается в разработке решения для автоматического детектирования курьеров доставки еды. Это решение имеет несколько ключевых целей: помочь ресторанам формировать общую оценку спроса на услуги доставки, предоставить градостроителям и правовым регуляторам данные для улучшения городской инфраструктуры, а также содействовать созданию эффективных законов и нормативных актов. Детектирование курьеров позволяет отслеживать их активность, анализировать маршруты и выявлять области с высоким уровнем спроса на доставку. Эти данные могут быть использованы для более рационального размещения ресторанов, оптимизации транспортных схем и разработки инициатив, направленных на улучшение городской среды.

Для решения задачи детектирования курьеров использовались нейронные сети. Одним из самых эффективных подходов является дообучение подготовленной модели на пользовательский класс [6][7]. Подобные решения способны обеспечить не только высокую скорость обучения и хорошее начальное приближение, но и удобство тренировки, благодаря отсутствующей необходимости в создании большого датасета. В связи с этим данный подход является одним из наиболее актуальных в настоящее время.

На сегодняшний день в крупных городах уже существует необходимая инфраструктура камер наблюдения, что позволяет использовать компьютерное зрение с целью улучшения качества жизни граждан, в связи с этим первостепенной задачей является создание моделей нейронных сетей, которые способны решать сложные и трудозатратные задачи [8].

II. Наборы данных

Для разработки и обучения модели детекции курьеров был создан собственный набор данных. Фотографии, использованные в этом датасете, были получены из открытых источников. В общей сложности в набор вошло 238 изображений, которые были собраны несколькими способами: скриншоты с видео на YouTube и изображения, полученные из поисковиков (рисунок 1).

Изображения из поиска по фотографиям: основной источник данных — поисковые системы, такие как Google Images и Yandex Images, откуда было собрано 145 изображений. Эти фотографии представляли курьеров в различных условиях: передвигающихся пешком или на велосипеде, с характерными рюкзаками за спиной, а также в различной униформе. На большинстве из этих изображений курьер был крупным планом и не на

транспортном средстве, что делает подобный набор не особо репрезентативным.

Скриншоты из видеороликов на YouTube. Для увеличения разнообразия набора данных были использованы 37 скриншотов, извлеченных из видеоматериалов, опубликованных на YouTube. Это позволило добавить в набор данных более реалистичные и динамичные сцены, включая кадры, снятые в движении и при разных углах съемки.

Важно было также включить изображения, на которых курьеры отсутствуют, чтобы модель могла корректно классифицировать и различать сцены. В набор данных вошло 56 таких изображений, включающих городские пейзажи, пешеходов без рюкзаков, автомобили и другие элементы, которые могли бы представлять потенциальные ложноположительные результаты (рисунок 2).

После сбора базового набора изображений произведено увеличение объема данных с использованием методов аугментации. Для каждой исходной фотографии было создано 5 аугментированных копий, что значительно расширило общий объем фотографий. Эти аугментированные изображения содержали в среднем 3-4 модификации, включая геометрические трансформации, изменение яркости и контрастности, наложение шума и размытия, а также применение цветовых фильтров [9]. Главным условием для включения полученного изображения в датасет было одинаковое количество меток в исходной и аугментированной версиях, подобное условие позволяет отсеять «бессмысленные» для обучения экземпляры, например, когда в новый box были выделены только ботинки курьера, по которым невозможно определить с достаточной точностью профессию человека.

Полученный датасет состоит более чем из 1400 изображений и демонстрирует различные позы и ситуации, в которых находятся курьеры на протяжении своей работы.

Для обучения модели датасет был разделен на обучающую и валидационную выборки в соотношении 80% к 20%. Это разделение было выполнено случайным образом, чтобы обеспечить равномерное распределение изображений разных типов в обеих выборках. В результате: обучающая выборка включала около 1150 изображений, которые использовались для оптимизации параметров модели. Валидационная выборка содержала около 250 изображений, которые применялись для оценки точности модели на данных, которые не использовались во время обучения. Такое соотношение позволяет эффективно использовать доступные данные, обеспечивая баланс между качеством обучения и возможностью проверки обобщающей способности модели. Включение негативных примеров в обе выборки также способствовало улучшению устойчивости модели к ошибкам второго рода.

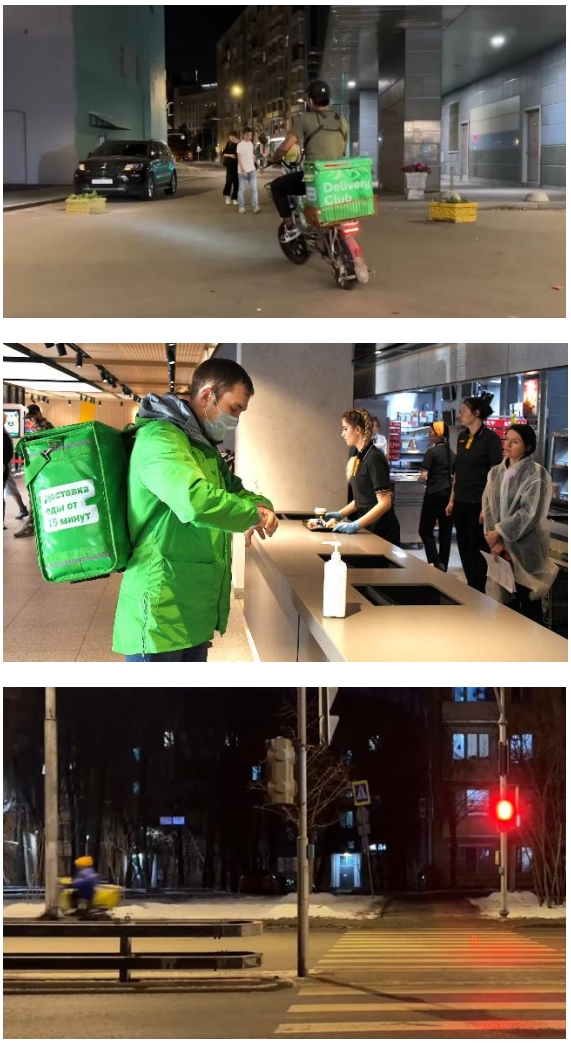


Рис. 1. Примеры изображений курьеров при различных обстоятельствах

Рис. 2. Примеры изображений (датасета), не включающие базовый класс

Аугментация была выполнена с использованием библиотеки Albumentations. Используя данный модуль, применялись следующие виды изменения изображения: GaussNoise, GaussianBlur, RandomBrightnessContrast, RandomGamma, ISONoise, ToGray, HueSaturationValue, BBoxSafeRandomCrop, Erasing, CoarseDropout и ShiftScaleRotate [10][11]. Большинство модификаций накладывались на изображения с вероятностью 0,5, однако для некоторых видов это значение было увеличено вплоть до 0,95. Так удалось добиться значительного увеличения вариативности набора данных (рисунок 3), что улучшило устойчивость модели.

Рис. 3. Пример оригинального изображения и случайных аугментаций

III. Нейросетевые архитектуры

YOLOv8 — это одна из последних версий популярной модели для детекции объектов в изображениях и видео, которая является частью семейства моделей YOLO (You Only Look Once). YOLOv8 обладает высокой производительностью, которая позволяет за миллисекунды определять классы на изображении. Также модель является достаточно гибкой и способна поддерживать различные типы задач, такие как классификация объектов, сегментация и детекция [12]. Модель семейства YOLO регулярно обновляется, что повышает не только их качество, но и актуальность. Также одним из преимуществ является единая платформа ultralytics, которая делает использование разных моделей удобным, дополнительно имеется возможность выполнять множество полезных функций, связанных с предобработкой исходных данных, например, изменение размера изображения и координатной разметки, а также дает удобную возможность гибкой настройки ключевых параметров обучения.

У YOLOv8 существует 5 ключевых видов предобученных моделей, которые отличаются масштабом и числом параметров. На каждый из этих видов есть своя версия нейронной сети, которая призвана решать определённый тип задач, такие как классификация, детекция, сегментация или работа с позами. Так модели, в которых после версии идет буква n, являются самыми маленькими и имеют наименьшее число обучаемых параметров и слоев. Остальные виды можно расположить в порядке возрастания комплексности модели следующим рядом: n, s, m, l, x [13]. Самая минимальная версия для детекции, обученная на данных COCO, оперирует 3.2 миллионов параметров, а самая большая — 68 миллионами, естественно, что от числа параметров зависят сложность обучения и скорость работы нейронной сети (рисунок 4).

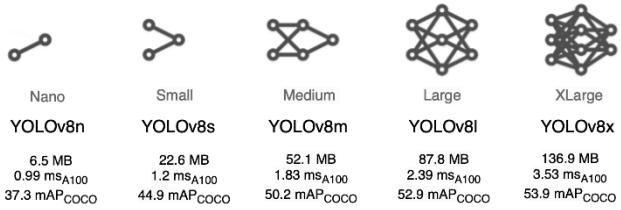


Рис. 4. Описание архитектурных особенностей моделей YOLOv8 для детекции

IV. Сравнение

Были обучены две версии модели: YOLOv8n и YOLOv8l. Обучение первой версии происходило в 50 эпох, полученная на выходе модель имела время обработки одного кадра около 2мс. Большая модель обучалась в течение 100 эпох, а время, требуемое на обработку одного кадра, увеличилось до 9мс. Обе нейронные сети показали быструю скорость работы, что позволяет их использовать с камерами, записывающие в 60 fps. Эти и последующие результаты оценки производительности проводились с использованием ROCM 6.2 [14]. В связи с нестабильностью работы нейронных сетей на видеокартах производства AMD данные могут быть нерепрезентативными (хуже, чем есть на самом деле).

Результаты обеих нейронных сетей были протестированы на тестовом видео, в котором изображены различные курьеры с разнообразных ракурсов. В разметке, полученной в результате обработки видео YOLOv8n, было много ошибок второго рода. Обучение YOLOv8l, прошло достаточно удачно, модель хорошо предсказывала курьеров, на видео практически нет ложных срабатываний, однако не все экземпляры базового класса были обнаружены, это может свидетельствовать, что часть ракурсов и поз доставщиков не находились в тренировочном датасете вообще или в достаточном количестве. Также стоит отметить, что на другом тестовом видео, нейросеть иногда определяла курьера в обычном челове-

ке. Данная проблема может возникать из-за дисбаланса классов в тренировочном датасете и недостаточного числа негативных примеров. Вероятно, если добавить в датасет недостающие изображения, то модель будет достаточно эффективной для всесторонней детекции курьеров, сейчас же хороший уровень уверенности достигается при обнаружении доставщика на средней и дальней дистанциях (рисунок 5).

Несмотря на применение аугментации с затиранием части изображения и обрезкой bbox, обе модели не смогли обработать один видеоряд, где нижняя часть курьера была частично закрыта забором, ни одна из нейронных сетей ни разу не выделила в данных кадрах курьера. Вероятно, данный эффект обосновывается проблемами с тренировочным датасетом, в котором было небольшое число похожих примеров, что не позволило нейронной сети полноценно произвести генерализацию и применить полученные результаты на реальном примере (рисунок 6).

Рис. 5. Разметка кадра из видео с использованием модели на базе YOLOv8l

Рис. 6. Один из кадров видеоряда, который не разметила ни одна из моделей

Также качество модели можно оценить, используя тестовые данные и метрики, которые предоставляет YOLO. Во время обучения PyTorch пытался минимизировать ошибки по следующим функциям потерь: box loss, cls loss и dfl loss.

box_loss — это функция потерь для регрессии ограничивающих рамок, которая измеряет ошибку в предсказанных координатах и размерах рамки по сравнению с истинными значениями. Чем ниже box_loss, тем точнее модель определяет положение и размеры объектов [15].

cls_loss — это функция потерь для классификации, которая измеряет ошибку в предсказанных вероятностях классов для каждого объекта на изображении по сравнению с истинными метками. Чем ниже значение cls_loss, тем точнее модель определяет классы объектов.

dfl_loss — distribution focal loss строится на основе фокальной функции потерь и используется для более точного предсказания координат ограничивающих рамок. Вместо предсказания фиксированных значений координат модель предсказывает распределение вероятностей вокруг каждого значения. Это позволяет улучшить регрессию рамок и повысить точность обнаружения объектов, особенно в сложных сценариях [16][17].

Для валидации полученной модели используются альтернативные методы: mAP50 и mAP50-95. Обе метрики являются наиболее репрезентативными и показывают пересечение предсказанного бокса с размеченным [18].

mAP50 представляет из себя среднюю точность предсказаний (mean average precision), рассчитанную при пороге пересечения рамок (IoU) 0,50. Этот показатель оценивает точность модели, учитывая только «простые» случаи обнаружения объектов.

mAP50-95 является средним значением mAP, рассчитанным при различных порогах IoU от 0,50 до 0,95. Этот показатель даёт более полное представление о качестве работы модели на объектах с разной степенью сложности обнаружения.

Исходя из полученных графиков, можно утверждать, что на протяжении всех 100 эпох нейронная сеть стабильно улучшала качество предсказания, возможно, что использование большего числа итераций улучшит результат, однако появляется вероятность переобучить модель, что наоборот ухудшит качество нейронной сети (рисунок 7).

Также YOLO позволяет посмотреть на полученный результат после каждой эпохи, чтобы визуально оценить работу модели (рисунок 8). Так, например, можно увидеть, что модель крайне неплохо справляется с тестовыми данными на примере одного из валидационных батчей. Об этом также свидетельствуют высокие показатели mAP.

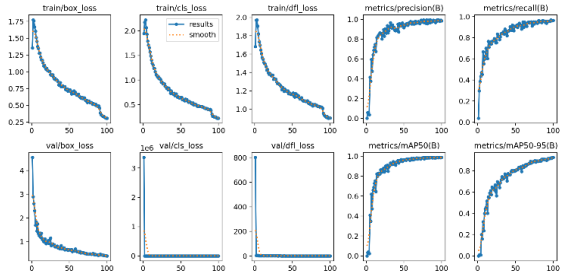


Рис. 7. Итоговые метрики процесса обучения для модели YOLOv8l

Рис. 8. Разметка тестовых изображений с использованием модели на базе YOLOv8l

V. Заключение

По результатам работы можно утверждать, что разметка курьеров в реальном времени для анализа и сбора статистики возможна, но требует качественного датасета. Также стоит рассмотреть потенциал использования более быстрых и маленьких версий YOLOv8, что теоретически поможет ускорить подбор наилучших гиперпараметров и работу анализа изображения без существенной потери в качестве.

Модели класса YOLO продемонстрировали себя как хороший выбор, но тем не менее для получения лучшего результата требуется сравнение с аналогами, например, YOLOv11, VGG19 или ResNet50. Но стоит учитывать, что одним из дополнительных преимуществ YOLO является ее постоянное развитие и удобство работы, что делает этот выбор на данный момент наиболее перспективным [19].

По результатам полученные модели могут использоваться с целью определения числа курьеров, проезжающих вдоль улицы, что поможет оценить нагрузку на общий поток, подтвердить или опровергнуть корреляцию со спросом по часам, а также помочь с решением о создании велодорожки.

В качестве потенциальных путей развития проекта можно выделить обновление и дополнение исходного датасета, более качественный подбор гиперпараметров и добавление дополнительных видов аугментации данных, например, изменение перспективы. После совершения указанных действий модель должна увеличить список ситуаций, в которых детекция будет осуществлена, а также убрать редкие ошибки второго рода.

Литература

[1] Иванихина А.А., Золоторева М.В. Нейросети и искусственный интеллект в сфере градостроительства // МОЛОДЁЖНЫЙ ВЕСТНИК НОВОРОССИЙСКОГО ФИЛИАЛА БЕЛГОРОДСКОГО ГОСУДАРСТВЕННОГО ТЕХНОЛОГИЧЕСКОГО УНИВЕРСИТЕТА ИМ. В. Г. ШУХОВА. - 2024. - №3 (15). - С. 16-22.

[2] Д.В. ГАВРИКОВ, В.Е. КАРПЕНКО ИСКУССТВЕННЫЙ ИНТЕЛЕКТ В ПРОЕКТИРОВАНИИ И ЭКСПЛУАТАЦИИ УСТОЙЧИВОЙ АРХИТЕКТУРЫ // АРХИТЕКТУРА И ДИЗАЙН: ИСТОРИЯ, ТЕОРИЯ, ИННОВАЦИИ. - 2024. - №8. - С. 306-312.

[3] А. А. Артамонов, Д. Ш. Дашкин, Е. А. Пекло Автоматизация системы мониторинга уличного движения с применением компьютерного зрения // МОЛОДЕЖНАЯ ШКОЛА-СЕМИНАР ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ В ТЕХНИЧЕСКИХ СИСТЕМАХ ИМЕНИ А.А. ВАВИЛОВА. - 2024. - №1. - С. 16-19.

[4] КУЗНЕЦОВА В.М., СЕРГЕЕВА С.М., БОГАТЫРЕВА Е.В. ПЕРСПЕКТИВЫ РАЗВИТИЯ СЕРВИСА ДОСТАВКИ В ИНДУСТРИИ ПИТАНИЯ // ИННОВАЦИОННОЕ РАЗВИТИЕ ТЕХНИКИ И ТЕХНОЛОГИЙ В ПРОМЫШЛЕННОСТИ (ИНТЕКС-2021) Сборник материалов Всероссийской научной конференции молодых исследователей с международным участием. Том Часть 6.. - М.: Федеральное государственное бюджетное образовательное учреждение высшего образования "Российский государственный университет имени А.Н. Косыгина (Технологии. Дизайн. Искусство), 2021. - С. 122-125.

[5] ИРИНИНА О.И. ТРЕНДЫ РАЗВИТИЯ РЕСТОРАННОГО БИЗНЕСА В 2024 ГОДУ // АКТУАЛЬНЫЕ АСПЕКТЫ ТЕОРИИ И ПРАКТИКИ РАЗВИТИЯ ИНДУСТРИИ ТУРИЗМА, ГОСТЕПРИИМСТВА И СЕРВИСА Материалы IV Международной научно-практической конференции.. - Владимир: Транзит-Икс, 2024. - С. 128-140.

[6] АЛИ Б., САДЕКОВ Р.Н., ЦОДОКОВА В.В. АЛГОРИТМЫ ИДЕНТИФИКАЦИИ СВЕТОФОРОВ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ В МУЛЬТИКАМЕРНЫХ СИСТЕМАХ ПОМОЩИ ВОДИТЕЛЮ // ГИРОСКОПИЯ И НАВИГАЦИЯ. - 2022. - №4 (119). - С. 87-105.

[7] В. О. Кирвяков ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ПРОМЫШЛЕННЫХ, КОММЕРЧЕСКИХ, МЕДИЦИНСКИХ И ФИНАНСОВЫХ ПРИЛОЖЕНИЯХ // сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики" Выпуск 2. - М.: Национальный исследовательский технологический университет "МИСИС", 2024. - С. 65-70.

[8] ЛОПУХОВ В.В. СОВРЕМЕННЫЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ВИДЕОАНАЛИТИКИ В РАСКРЫТИИ ПРЕСТУПЛЕНИЙ // ВЕСТНИК ВСЕРОССИЙСКОГО ИНСТИТУТА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ СОТРУДНИКОВ МИНИСТЕРСТВА ВНУТРЕННИХ ДЕЛ РОССИЙСКОЙ ФЕДЕРАЦИИ. - 2024. - №1 (69). - С. 122-127.

[9] ШУРШЕВ Т.В., ПУРЦМАН А.А., ЕЛЬЧАНИНОВА К.А. МЕТОД АУГМЕНТАЦИИ ДАННЫХ КАК ИНСТРУМЕНТ ДЛЯ РЕШЕНИЯ ПРОБЛЕМЫ НЕХВАТКИ ДАННЫХ // НАУКА. ИННОВАЦИИ. БУДУЩЕЕ - 2024 Сборник статей Международной научно-практической конференции.. - Петрозаводск: Международный центр научного партнерства «Новая Наука» (ИП Ивановская И.И.), 2024. - С. 65-70.

[10] Albumentations Documentation // Albumentations URL: https://albumentations.ai/docs/ (дата обращения: 01.07.2025).

[11] Albumentations // Read The Docs URL: https://albumentations.readthedocs.io/en/latest/ (дата обращения: 01.07.2025).

[12] Efficient Object Detection with YOLOV8 and KerasCV // Keras URL: https://keras.io/examples/vision/yolov8/ (дата обращения: 07.01.2025).

[13] Ultralytics YOLOv8 // ultralytics URL: https://docs.ultralytics.com/models/yolov8/ (дата обращения: 07.01.2025).

[14] Installing PyTorch for ROCm // AMD URL: https://rocm.docs.amd.com/projects/install-on-linux/en/docs-6.2.0/install/3rd-party/pytorch-install.html (дата обращения: 07.01.2025).

[15] What is box loss in yolov8? // YOLOv8 URL: https://yolov8.org/what-is-box-loss-in-yolov8/#The_Mathematics_Behind_Box_Loss (дата обращения: 07.01.2025).

[16] Reference for ultralytics/utils/loss.py // ultralitics URL: https://docs.ultralytics.com/reference/utils/loss/ (дата обращения: 07.01.2025).

[17] YOLO Loss Function Part 2: GFL and VFL Loss // LearnOpenCV URL: https://learnopencv.com/yolo-loss-function-gfl-vfl-loss/ (дата обращения: 07.01.2025).

[18] Performance Metrics Deep Dive // ultralitics URL: https://docs.ultralytics.com/guides/yolo-performance-metrics/ (дата обращения: 07.01.2025).

[19] ГУЖВА Н.С., САДЕКОВ Р.Н. АЛГОРИТМЫ ИДЕНТИФИКАЦИИ СВЕТОФОРОВ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ В МУЛЬТИКАМЕРНЫХ СИСТЕМАХ ПОМОЩИ ВОДИТЕЛЮ // ГИРОСКОПИЯ И НАВИГАЦИЯ. - 2024. - №3 (126). - С. 47-65.

[20] Матяш Д.С. Детекция беспилотных летательных аппаратов на фотографиях с использованием методов компьютерного зрения // сборник статей научно-технического семинара студентов кафедры "Инженерной кибернетики" Выпуск 2. - М.: Национальный исследовательский технологический университет "МИСИС", 2024. - С. 92-99.

ГРНТИ 20.53.19
ASJC 1707
OECD 1.02 EP

Обнаружение строительных касок на рабочих для обеспечения безопасности в реальных условиях

Д. В. Шахов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
hunturek@edu.misis.ru

***Аннотация* — безопасность на строительных площадках является важной задачей, требующей постоянного контроля за состоянием работников. Одним из ключевых факторов безопасности является ношение касок. В данной работе рассматривается использование методов компьютерного зрения для автоматической детекции касок на строителях с целью повышения безопасности в условиях, приближенных к реальности. Для решения задачи применена модель YOLOv8, обученная на специально подготовленном датасете, включающем изображения строителей в касках и без касок. Оценка производительности модели показала высокие результаты для распознавания касок и людей, что подтверждает эффективность предложенного подхода для обеспечения безопасности на строительных площадках.**

Ключевые слова — компьютерное зрение, детекция касок, распознавание касок, безопасность, безопасность на стройплощадке, YOLO, mAP, COCO

I. Введение

Изучение и применение технологий компьютерного зрения в различных областях, включая промышленность и безопасность, активно развивается в последнее десятилетие. Особое внимание уделяется задачам детектирования и распознавания объектов в реальном времени, включая задачи, связанные с охраной труда и строительством. Важным аспектом является автоматическое определение наличия защитного снаряжения, в том числе касок, у работников строительных объектов, что может значительно повысить уровень безопасности на производстве. Также отмечено, что подавляющее большинство, а именно порядка 63% несчастных случаев, происходит из-за пренебрежения работниками средствами индивидуальной защиты. Поэтому имеет смысл реализовывать такие системы контроля, которые могли бы контролировать ношение тех или иных элементов средств индивидуальной защиты (СИЗ) [1].

Методы глубокого обучения показали высокую производительность и способность к обобщению в задачах данного типа — особенно таких как обнаружение и классификация [2]. Для решения этих задач активно используются методы компьютерного зрения, особенно с применением нейросетевых архитектур глубокого обучения. Одной из ключевых задач является точное и быстрое обнаружение касок на изображениях, что требует высокой точности и минимизации ложных срабатываний [3]. С развитием глубокого обучения и нейросетевых технологий методы детектирования человека достигли значительных успехов. Современные архитектуры нейронных сетей способны обрабатывать огромные объемы данных, извлекая из них значимые признаки, которые позволяют с высокой точностью определять местоположение и идентичность людей в изображениях и видео [4].

Существуют каски общего назначения, которые используют в разных отраслях. Каски разделяют по цвету, указывающему на должностную принадлежность работника, но расцветка защитной каски в стандартах не регламентируется — решение о выборе цвета принимается организациями самостоятельно [5]. В настоящее время для решения подобных задач широко применяются методы детекции объектов, такие как YOLO и Faster R-CNN, которые позволяют эффективно распознавать объекты на изображениях. Однако для корректного детектирования строительных касок эти модели необходимо дообучить на специализированных данных.

В настоящее время для решения таких задач широко применяются методы детекции объектов, такие как YOLO и Faster R-CNN [6], которые позволяют эффективно распознавать объекты, включая каски, на изображениях. В данной статье рассматриваются достижения в области применения глубокого обучения для распознавания касок у работников строительных объектов на изображениях и видеопотоках.

II. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались некоторые наборы данных, как локальные, собранные авторами, так и открытые. Рассмотрим используемые открытые наборы.

A. COCO

COCO (Common Objects in Context) — это один из самых популярных и широко используемых датасетов в области компьютерного зрения, предназначенный для задач обнаружения и сегментации объектов. COCO включает более 330 000 изображений, на которых размечены более 80 категорий объектов. Эти категории охватывают широкий спектр объектов, таких как люди, транспортные средства, животные, еда и другие.

Особенность COCO заключается в том, что датасет включает изображения с различными условиями освещения, разнообразными ракурсами и сложными сценами, что позволяет тренировать модели, способные эф-

фективно справляться с реальными задачами детектирования. Например, на изображениях могут быть люди в разных позах, частично скрытые или на дальнем фоне, что повышает сложность задачи.

Для задачи детектирования людей, как и для других объектов, COCO предоставляет аннотации, которые могут быть использованы для обучения моделей. Однако для специфичных задач, таких как распознавание касок на людях, может потребоваться использование более специализированных датасетов, которые учитывают именно эти особенности. Важным аспектом является то, что COCO предоставляет разнообразие сложных сценариев, включая случаи с низким освещением или частичным покрытием объектов, что помогает в обучении более универсальных детекторов.

Рис. 1. Примеры изображений с людьми, взятые из COCO

Рис. 2. Сложные случаи для детектирования:
а) низкое освещение; б) человек с частичным покрытием; в) нестандартный ракурс; г) человек в движении

B. Пользовательский датасет

Для решения задачи детектирования касок на людях был собран пользовательский датасет, состоящий из изображений, взятых из интернета. Изначально датасет включал 350 изображений, на которых присутствовали как люди в касках, так и без касок. Для улучшения качества обучения и повышения разнообразия данных была проведена аугментация, которая включала следующие методы:

- Автоматическая ориентация пиксельных данных с удалением EXIF-ориентации, чтобы изображения были приведены к единому виду.
- Изменение размера изображений до 640x640 пикселей (с растяжением изображения).

Кроме того, для создания трех версий каждого исходного изображения были применены методы аугментации:

- Равная вероятность одного из следующих поворотов на 90 градусов: без поворота, по часовой стрелке, против часовой стрелки, перевернутое изображение.
- Случайный поворот изображения в диапазоне от -15 до +15 градусов.
- Случайный сдвиг изображения в пределах от -15° до +15° по горизонтали и вертикали.

После применения методов аугментации количество изображений в датасете увеличилось до 1100, что позволило создать более разнообразный и обогащенный набор данных. Для каждой картинки была произведена разметка объектов на 3 класса: person (человек), helmet (каска) и person_without_helmet (человек без каски). Разметка была выполнена с использованием прямоугольных ограничивающих рамок (bounding boxes), что является стандартным методом аннотирования объектов в задачах детектирования.

Этот датасет представляет собой ценное дополнение к существующим базам данных, обеспечивая возможность решения специфической задачи распознавания касок на людях в различных условиях. Всего аннотировано более 5 тысяч объектов различных классов.

Рис. 3. Аннотация изображений на платформе CVAT

Рис. 4. Пример искажения изображения методами аугментации

III. Нейросетевые архитектуры

YOLOv8 является одной из последних модификаций популярной модели YOLO, предназначенной для задач детектирования объектов. Модель YOLOv8 использует модификацию архитектуры CSPDarknet53, которая состоит из 53 сверточных слоев [6]. Эта архитектура применяет частичные межэтапные соединения, что улучшает поток информации между различными уровнями сети и способствует повышению точности распознавания объектов. Модель делится на две основные части: основную часть, отвечающую за извлечение признаков, и "голову", которая прогнозирует ограничивающие рамки и классы объектов [7].

Принцип работы архитектуры YOLO заключается в том, что входное изображение разделяется на равные ячейки сетки размером $S \times S$. Каждая ячейка сети отвечает за предсказание объектов, прогнозируя B ограничивающих рамок и соответствующие им оценки уверенности (вероятности).

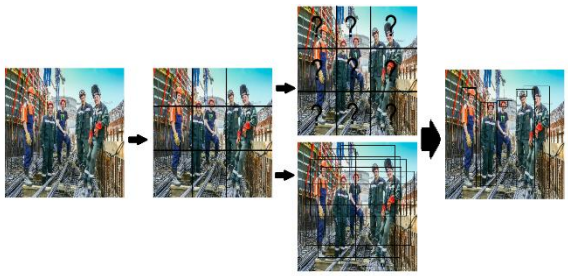


Рис. 5. Принцип работы архитектуры YOLO

Для решения проблемы перекрытия рамок используется метод, основанный на показателе пересечения (IoU), который определяет, какие рамки являются наиболее подходящими для локализации объекта. Высокое значение IoU указывает на точное совпадение между предсказанными и реальными рамками, что важно для повышения точности модели [8].

Особенностью YOLOv8 является использование различных техник аугментации изображений в процессе обучения, что позволяет улучшить обобщающие способности модели. Одной из таких техник является мозаичная аугментация, при которой несколько изображений объединяются в одно. Это позволяет улучшить детектирование объектов, расположенных в различных частях изображения и на фоне, что особенно полезно при обработке изображений с неполным отображением объектов, таких как каски.

Для оценки качества работы модели применяется метрика mAP (mean Average Precision), которая является стандартом для задач детектирования объектов. YOLOv8 достигает высокой точности на COCO. Например, модель YOLOv8m — средняя модель — достигает 50,2% mAP при измерении на COCO. Всего существует 5 вариантов модели YOLOv8 (таблица 1) [6,7].

При тестировании на датасете COCO, модель YOLOv8x показала самый высокий результат по данной метрике, что демонстрирует более высокую точность модели по сравнению с младшими. Эта модель была выбрана для дообучения в рамках текущей работы, так как она является наиболее точной, что особенно важно для задачи детектирования касок, где требуется высокая надежность и минимизация ложных срабатываний. В процессе дообучения сохраняется большинство параметров предварительно обученной модели, что ускоряет обучение и позволяет эффективно адаптировать модель под специфические задачи, такие как распознавание касок.

ТАБЛИЦА 1. Вариации модели YOLOv8.

Модель	Размер изображения	mAP(50-95)	Скорость A100 TensorRT (мс)	Параметры (M)
YOLOv8n	640	37.3	0.99	8.7
YOLOv8s	640	44.9	1.2	28.6
YOLOv8m	640	50.2	1.83	78.9
YOLOv8l	640	52.9	2.39	165.2
YOLOv8x	640	53.9	3.53	257.8

IV. Оценка метрик

Для оценки качества распознавания объектов применяют три метрики: точность (Precision), которая показывает долю верно предсказанных объектов, полноту (Recall), отражающую долю верно найденных объектов среди всех объектов, а также mAP (mean Average Precision), представляющую собой среднее значение Average Precision для каждого класса, что эквивалентно площади под кривой Precision-Recall.

$$precision = \frac{\mathrm{TP(c)}}{\mathrm{TP(c)} + \mathrm{FP(c)}}$$

$$recall = \frac{\mathrm{TP(c)}}{\mathrm{TP(c)} + \mathrm{FN(c)}}$$

где $TP(c)$ - количество предсказаний True Positive для класса c , где $FP(c)$ - количество предсказаний False Positive для класса c, где $FN(c)$ - количество предсказаний False Negative для класса c.

Для практической части была дообучена модель YOLOv8x с использованием фреймворка Ultralytics версии 8.3.55. В процессе обучения использовались следующие параметры: модель `yolov8x.pt`, задачи детектирования (`task=detect`), количество эпох — 50, размер изображения — 640x640 пикселей, размер батча — 16, использовался оптимизатор по умолчанию. Обучение проводилось с использованием GPU Tesla P100. Модель была дообучена с предварительно обученными весами (`pretrained=True`). Для оптимизации использовался коэффициент моментума 0.937 и декремент скорости обучения 0.01. В процессе обучения была

включена аугментация данных с использованием мозаики, изменений масштаба, поворотов, а также применения случайных сдвигов и переворотов изображений. Для проверки качества модели использовался стандартный коэффициент IoU 0.7 и количество детекций на изображении ограничивалось 300 объектами.

Полученные метрики лучшей из получившихся в результате обучения моделей (49 эпоха) приведены в таблице 2.

ТАБЛИЦА 2. Результаты обучения.

Class	P	R	mAP50	mAP50-95
all	0.851	0.714	0.816	0.512
helmet	0.953	0.749	0.885	0.535
person	0.892	0.724	0.855	0.562
person_without_helmet	0.706	0.669	0.708	0.439

На рисунке 6 представлены графики зависимости метрик и ошибок от количества эпох обучения, построенные на тестовом наборе данных. Из графиков видно, что значения метрик продолжают расти, а ошибки не увеличиваются, что свидетельствует о том, что модель не переобучается. Однако такой тренд также указывает на то, что количество эпох обучения может быть недостаточным для достижения оптимальных результатов. Продолжение обучения потребует значительных вычислительных ресурсов и времени.

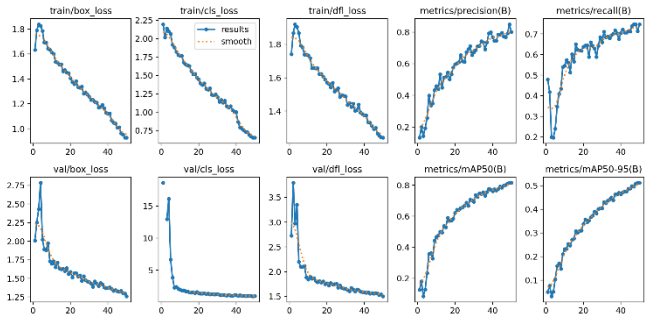


Рис. 6. – Зависимость значений метрик и ошибок от эпох в процессе обучения модели.

Более детальная статистика отображена в нормализованной матрице ошибок на рисунке 7. В контексте задачи распознавания объектов, такая матрица показывает, как предсказания модели соотносятся с истинными метками классов.

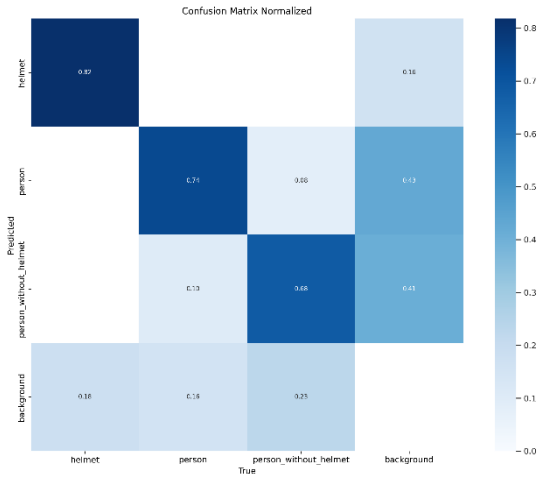


Рис. 7. – Нормализованная матрица ошибок.

Результат работы дообученной нейросети YOLOv8x представлен на рисунке 8:

Рис. 8. – Результаты предсказаний модели.

Результаты обучения модели демонстрируют, что значения ключевых метрик, таких как точность (Precision), полнота (Recall) и средняя точность (mAP), сопоставимы с результатами, достигнутыми моделями YOLO на датасете COCO — признанном эталоне в задачах детекции объектов. Это указывает на то, что модель успешно справляется с задачей обнаружения и

классификации объектов, таких как человек, защитная каска и человек без каски, в рамках предоставленного набора данных. Сопоставимость метрик позволяет утверждать, что модель обладает высокой точностью и обобщающей способностью для решения поставленной задачи.

V. Заключение

Были рассмотрены основные наборы данных, на которых обучались и тестировались рассматриваемые нейронные сети. Подробно рассмотрен пользовательский датасет, разобран принцип работы применяемой модели и выбрана оптимальная для данной задачи версия.

В третьей части была произведена оценка ключевых метрик дообученной модели и был подведен итог: в связи с сопоставимостью полученных метрик с метриками предобучения YOLOv8x на датасете COCO, получившуюся модель можно считать хорошо справляющейся с поставленной задачей детектирования строительных касок на рабочих в условиях, приближенных к реальным.

Литература

[1] Фирсов, О. А. Разработка системы распознавания касок, с помощью архитектуры YOLOv5 / О. А. Фирсов // Современные проблемы горно-металлургического комплекса. Наука и производство : Материалы девятнадцатой Всероссийской научно-практической конференции с международным участием, Старый Оскол, 07 декабря 2022 года. – Старый Оскол: Национальный исследовательский технологический университет "МИСиС", 2023. – С. 540-543. – EDN UPBBOM.

[2] Антипов И. И. Исследование возможности классификации мусора при помощи компьютерного зрения // Сборник статей научно-технического семинара студентов кафедры «инженерной кибернетики» на тему «искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях» // Кафедра инженерной кибернетики, НИТУ «МИСиС», 2023. – С. 16-21.

[3] Использование нейросетевых детекторов для предотвращения несчастных случаев на производстве / В. А. Егунов, П. Д. Кравченя, Е. И. Большакова, З. Ш. Сутенова // Инженерный вестник Дона. – 2023. – № 10(106). – С. 39-47. – EDN DPEGBS.

[4] Карякин А. В. Исследование задачи детектирования человека с помощью компьютерного зрения // Сборник статей научно-технического семинара студентов кафедры «инженерной кибернетики» на тему «искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях» // Кафедра инженерной кибернетики, НИТУ «МИСиС» , 2024. – С. 61-64.

[5] Нечаева, А. В. Азработка системы контроля наличия касок на работниках предприятий с помощью методов машинного зрения / А. В. Нечаева, Ю. А. Цыганков, Д. А. Полещенко // Современные проблемы горно-металлургического комплекса. Наука и производство : Материалы девятнадцатой Всероссийской научно-практической конференции с международным участием, Старый Оскол, 07 декабря 2022 года. – Старый Оскол: Национальный исследовательский технологический университет "МИСиС", 2023. – С. 434-439. – EDN ELUPHF.

[6] Л. С. Толстенко, А. А. Клейменов, Б. Али [и др.], Анализ нейронных сетей для детектирования светофоров на изображениях // Научно-технический журнал «Известия Института Инженерной Физики», 2023 (№2) – С. 59-65.

[7] Рамзайцев Д. А., Матяш Д. С. Исследование возможности распознавания объектов на спутниковых снимках // Сборник статей научно-технического семинара студентов кафедры «инженерной кибернетики» на тему «искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях» // Кафедра инженерной кибернетики, НИТУ «МИСиС», 2023. – С. 127-132.

[8] “Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection” available at https://www.researchgate.net/publication/342027416_Generalized_Fo cal_Loss_Learning_Qualified_and_Distributed_Bounding_Boxes_for _Dense_Object_Detection (Accessed December 19, 2023) Behrendt K., Novak L., Botros R. “A deep learning approach to traffic lights: Detection, tracking, and classification”, 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1370–1377.