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 СТУДЕНТОВ КАФЕДРЫ «ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ» НА ТЕМУ «ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ПРОМЫШЛЕННЫХ, КОММЕРЧЕСКИХ, МЕДИЦИНСКИХ И ФИНАНСОВЫХ ПРИЛОЖЕНИЯХ»

Москва 2024

УДК 004.8
ББК 32.813.5

Искусственный интеллект в промышленных, коммерческих, медицинских и финансовых приложениях, 2024: Сборник статей научно-технического семинара студентов. Вып. 2 / Под ред. А.Р. Ефимова— М.: НИТУ «МИСИС», 2024.— 152 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 30-31 мая 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 28.23.37
ASJC 1707
OECD 1.02

Определение положения тела человека с использованием нейронных сетей

А. А. Абакумов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2305400@edu.misis.ru

В. О. Хуако
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
v.khuako@edu.misis.ru

***Аннотация* — В данной статье рассматривается актуальная задача определения положения тела (pose detection) в области компьютерного зрения (CV). Для её решения применяется метод локализации ключевых точек тела (суставов). В работе проводится анализ двух решений с открытым исходным кодом: OpenPose и Kapao. Сравнивается их эффективность в определении положения тела человека на изображениях. Для исследования используются изображения из открытых наборов данных: COCO и MPII.**

Ключевые слова —Компьютерное зрение, Распознавание положения тела, OpenPose, Kapao, COCO, MPII.

I. Введение

В настоящее время, в свете стремительного развития технологий, большое внимание уделяется исследованиям в области компьютерного зрения. Данные исследования охватывают широкий спектр вопросов, таких как: разработка навигационных систем [1], разработка алгоритмов для автономной коррекции навигационных систем на основе распознавания дорожной и речной сети [2], визуальная локализация наземных транспортных средств с помощью монокамер и дорожных знаков с геодезическими границами [3], использование нейронных сетей для распознавания светофоров на изображениях [4], повышение точности сопровождения подвижных объектов с помощью алгоритма комплексной обработки сигналов с монокулярной камеры и LIDAR [5] и др.

В данной статье рассмотрим одно из направлений в области компьютерного зрения, а именно определение положения тела человека, что состоит из распознавания и локализации частей тела на изображениях и видео [6]. Над решением задач определения положения тела человека работают университеты [7], крупные технологических компании [8] и малые стартапы [9].

Технология распознавания положения тела находит своё применение в различных областях, представим некоторые из них:

- В VR и AR позволяет пользователям взаимодействовать с виртуальным окружением [9].
- В медицине позволяет отслеживать процесс реабилитации пациентов путем анализа их движений для оценки прогресса лечения [10].
- В сфере безопасности распознавание подозрительных действия системами видеонаблюдения [11].
- Управление интерфейсами и промышленными роботами на основе жестов [9].

Определение положения тела на видео и изображениях исследуются уже несколько десятилетий и существуют несколько решений данной задачи:

- Классические алгоритмы основаные на использовании вручную настраиваемых признаков и алгоритмов обнаружения объектов, таких как детекторы краев, дескрипторы углов или каскады Хаара. Эти методы часто требуют предварительной обработки изображений и настройки параметров для конкретной задачи [12].
- Системы маркеров и захвата движений, эти системы используют специальные маркеры или носимые устройства, чтобы отслеживать движения человека в реальном времени. Такие системы обычно используются в киноиндустрии, анимации, спортивных тренировках и научных исследованиях [13].
- Методы, основанные на машинном обучении, что в последнее время набирают всё большую популярность, такие как CNN [14] (Convolutional Neural Networks), что могут автоматически извлекать признаки из изображений и видео, что делает их более эффективными и универсальными.

II. Наборы данных

В рамках проведения экспериментов и обучения анализируемых нейронных сетей использовались общедоступные наборы данных подробно проанализируем их.

A. MPII

MPII (max planck institute informatik) – набор данных созданный для оценки и обучения работы с распознаванием позы человека, используя методы нейронных сетей [15].

Набор данных предоставляет около 25 тыс. изображений, содержащих более 40 тыс. человек с аннотированными суставами тела и охватывает 410 видов человеческой деятельности, и каждое изображение имеет соответствующую метку деятельности.

Каждое изображение извлечено из видео на платформе YouTube и снабжено предшествующими и последующими неаннотированными кадрами. Тестовый набор изображений предоставляет подробные аннотации, включая окклюзии частей тела и 3D ориентацию торса и головы.

Примеры изображений входящих в набор данных MPII представлены на Рис. 1.

Рис. 1. Пример изображений из набора MPII

B. Microsoft COCO Keypoints

Рассмотрим COCO [16] (Common objects in context) - набор данных, что в отличии от MPII предоставляет изображения не только для определения положения тела, но и для задач обнаружения, сегментации объектов и создания надписей. Используется в машинном обучении для исследований и практического применения.

COCO содержит более 330 тыс. изображений, каждое аннотировано 80 категориями объектов и 5 подписями, описывающими сцену.

Набор состоит из двух частей: изображений и аннотаций к ним.

1. Изображения организованы в иерархию каталогов, причем каталог верхнего уровня содержит подкаталоги для обучающего, проверочного и тестового наборов.
2. Аннотации представлены в формате JSON, причем каждый файл соответствует одному изображению.

Каждая аннотация в наборе данных содержит следующую информацию:

- Имя файла изображения.
- Размер изображения (ширина и высота).
- Список объектов со следующей информацией: Класс объекта (например, "человек", "автомобиль"); Координаты ограничительной рамки (x, y, ширина, высота); Маска сегментации (полигон или формат RLE); Ключевые точки и их положение (если доступно).
- Пять подписей, описывающих сцену.

COCO предлагает различные типы аннотаций:

- Обнаружение объектов с координатами ограничительных рамок и полными масками сегментации для 80 различных объектов.
- Сегментация изображений "вещей" с помощью пиксельных карт, отображающих 91 аморфную фоновую область.
- Паноптическая сегментация определяет объекты на изображениях, основываясь на 80 категориях «вещей» и 91 категории «материала».
- Более чем 39 000 фотографий с более чем 56 000 помеченных человек с отображением пикселей и шаблона 3D-модели и описаний на естественном языке для каждого изображения.
- Аннотации ключевых точек для более чем 250 000 человек с указанием ключевых точек, таких как правый глаз, нос и левое бедро.

Пример изображений с аннотациями, содержащиеся в наборе данных COCO представлены на Рис. 2

Рис. 2. Примеры изображений с аннотациями ключевых точек в наборе данных COCO

C. CrowdPose

Набор данных CrowdPose [17] содержит в общей сложности 20 000 изображений, где изображены 80 000 человек с 14 помеченными ключевыми точками. Тестовый набор включает 8 000 изображений. Его индекс толпы удовлетворяет равномерному распределению в [0, 1]. Изображения скопления людей внутри помещений были получены из MSCOCO, MPII.

Примеры изображений CrowdPose представлены на Рис. 3.

Рис. 3. Пример изображений из набора CrowdPose

III. Нейросетевые архитектуры

A. *OpenPose*

Нейросеть OpenPose [18] (Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields) решает задачу определения положения тела человека на видео и изображениях. На Рис. 4 показана общая схема метода OpenPose.

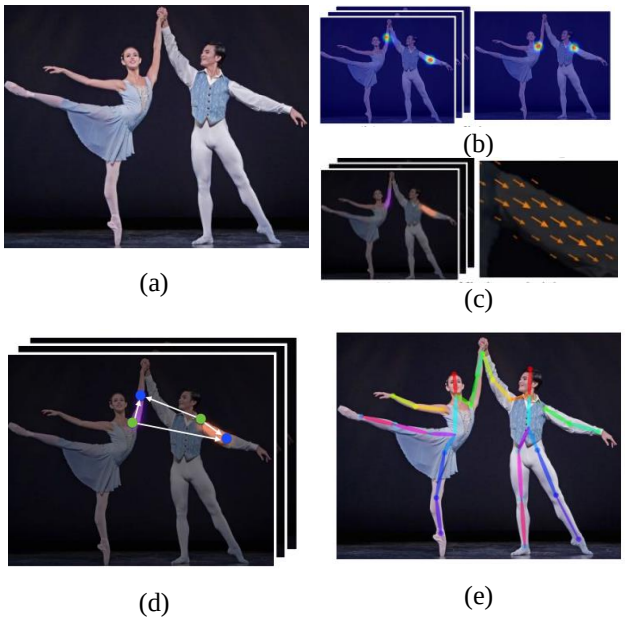


Рис. 4. общая схема метода OpenPose

Система принимает на вход цветное изображение размером w × h (Рис. 4a) и выдает двумерное расположение анатомических ключевых точек для каждого человека на изображении (Рис. 4e). Сначала сеть прямого распространения прогнозирует набор двумерных карт достоверности S расположения частей тела (Рис. 4b) и набор двумерных векторных полей L полей сродства частей (PAF), которые кодируют степень ассоциации между частями (Рис. 4c). Набор S = (S_1, S_2, ..., S_j) имеет J карт уверенности, по одной на каждую часть, где $S_j \in R^{w \times h}$, j ∈ {1 . . . J}. Множество L = (L_1, L_2, ..., L_C) имеет C векторных полей, по одному на конечность, где $L_C \in R^{w \times h \times 2}$, c ∈ {1 . . . C}. Далее, карты достоверности и PAF разбираются с помощью жадного вывода (Рис. 4d), чтобы вывести 2D-ключевые точки для всех людей на изображении.

Авторы OpenPose используют, как основу архитектуру многоступенчатой CNN, представленную на Рис. 5, что итеративно предсказывает поля сходства деталей (PAFs), которые кодируют ассоциацию между частями, показанные синим цветом, и карты достоверности обнаружения, показанные бежевым цветом. Архитектура итеративного предсказания, уточняет предсказания на последовательных этапах, t ∈ {1, . . ., T}, с промежуточным контролем на каждом этапе.

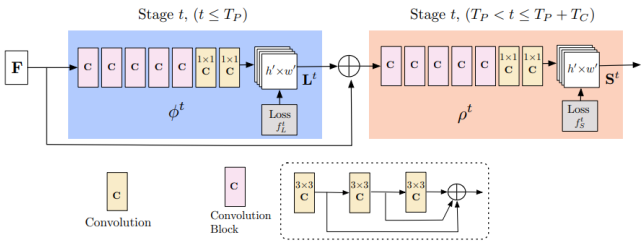


Рис. 5. архитектура многоступенчатой CNN

Первый набор этапов предсказывает PAF L^t , а последний - карты достоверности S^t . Предсказания каждого этапа и соответствующие им признаки изображения объединяются для каждого последующего этапа.

В оригинальном подходе архитектура сети включала несколько конволюционных слоев 7x7. В текущей модели рецептивное поле сохраняется, а вычисления сокращаются за счет замены каждого сверточного ядра 7x7 на 3 последовательных ядра 3x3. Если в первом случае количество операций составляет 2 × 72–1 = 97, то во втором - всего 51. Кроме того, выход каждого из трех сверточных ядер конкатенируется, следуя подходу, аналогичному DenseNet [19]. Количество слоев нелинейности увеличивается в три раза, и сеть может сохранять признаки как нижнего, так и верхнего уровня.

Одновременное обнаружение и ассоциация реализуется в OpenPose следующим образом: изображение анализируется CNN, генерирующей набор карт признаков, который является входом для первого этапа. На этом этапе сеть генерирует набор полей сродства частей (PAFs). На каждом последующем этапе прогнозы, полученные на предыдущем этапе, и исходные признаки изображения объединяются и используются для получения уточненных прогнозов. После всех этапов итераций процесс повторяется для определения карт достоверности, начиная с самого обновленного прогноза PAF.

Карта достоверности в OpenPose — это двумерное представление уверенности в то, что определенная часть тела может быть расположена в заданном пикселе. Если на изображении один человек, то на каждой карте достоверности должен быть один пик, если соответствующая часть тела видна; если на изображении несколько человек, то для каждого из них должен быть пик, соответствующий каждой видимой части.

Индивидуальные карты достоверности $S^*_{j,k}$создаются для каждого человека k. Пусть $x_{j,k} \in \mathbb{R}^2$ - истинное положение части тела j для человека k на изображении. Значение в месте P ∈ $\mathbb{R}^2$ в $S^*_{j,k}$определяется по формуле (1).

$$S^*_{j,k}(P) = \exp\left(-\frac{||P - x_{j,k}||_2^2}{\sigma^2}\right) \qquad (1)$$

где σ определяет разброс пика. Карта достоверности, предсказанная сетью, представляет собой объединение отдельных карт достоверности с помощью оператора max, по формуле (2).

$$S_j^*(P) = \max_k S_{j,k}^*(P) \quad (2)$$

OpenPose использует максимумы карт достоверности, а не среднее значение, чтобы точность близлежащих пиков оставалась различной, как показано на Рис. 6.

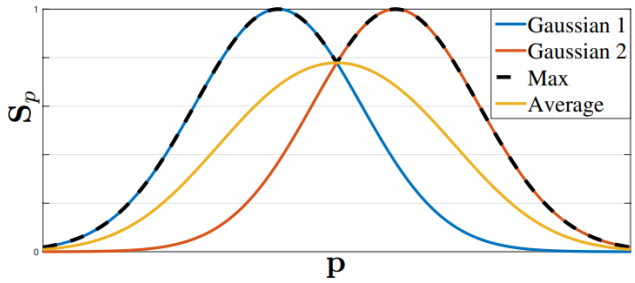


Рис. 6. карты достоверности

OpenPose для ассоциация частей тела использует Part Affinity Fields (PAF). Каждое PAF представляет собой двумерное векторное поле для каждой конечности. Для каждого пикселя в области, относящейся к определенной конечности, двумерный вектор кодирует направление, указывающее от одной части конечности к другой. Каждый тип конечности имеет соответствующий PAF, соединяющий две связанные с ним части тела.

Обучение OpenPose происходило на 2 наборах данных:

1. MPII
2. Microsoft COCO Keypoints

B. *Karao*

Рассмотрим Karao [20] (Keypoints And Poses As Objects), что решает задачу определения положения тела человека, где доминирующим подходом является регрессия на основе тепловых карт.

Karao использует плотную сеть обнаружения для одновременного предсказания набора объектов ключевых точек и набора объектов положения тела, вместе взятых.

1. Объект ключевой точки — это адаптация обычного представления объекта, в котором координаты ключевой точки представлены в центре небольшого ограничительного поля с равными шириной и высотой.
2. Объект положения тела рассматривается как расширение обычного представления объекта, которое дополнительно включает набор ключевых точек, связанных с объектом.

Оба представления объектов обладают уникальными преимуществами. Объекты ключевых точек специализированы для обнаружения отдельных ключевых точек, которые характеризуются ярко выраженными локальными особенностями. В отличие от этого, объекты положения тела лучше подходят для локализации ключевых точек со слабыми локальными признаками, поскольку они позволяют сети изучать пространственные отношения внутри набора ключевых точек.

Karao разработана для одновременного обнаружения обоих типов объектов с минимальными вычислительными затратами с использованием одной общей сетевой головки. В процессе вывода более точные обнаружения ключевых объектов объединяются с обнаружениями положения тела человека с помощью простого алгоритма согласования на основе допусков, который повышает точность предсказания положения тела человека без значительного снижения скорости вывода.

Архитектура Karao представленная на Рис.7 использует глубокую сверточную нейронную сеть (CNN) для сопоставления входного RGB-изображения с набором из четырех выходных сетей, содержащих прогнозы объектов, где N это количество якорных каналов и количество выходных каналов для каждого объекта.

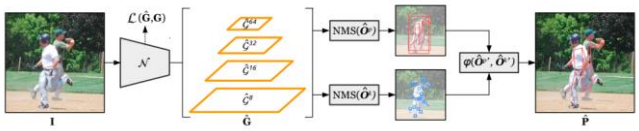


Рис. 7. Общая схема архитектуры Karao

Karao использует плотную сеть обнаружения, обученную с помощью функции потерь при многозадачности, для отображения RGB-изображения на набор выходных сетей, содержащих объекты с предсказанным положением тела и объекты с ключевыми точками.

Также Karao использует не максимальное подавление (NMS) [21] для получения кандидатов на обнаружение и, которые объединяются с помощью алгоритма согласования для получения окончательного прогноза положения тела человека.

Karao обучалась на двух наборах данных:

1. Microsoft COCO Keypoints, где входные изображения были изменены и дополнены до размера 1280×1280 и обучалась в течение 500 эпох.
2. CrowdPose, где обучение происходило на тренировочном сплите с 12 тыс. изображений и тестировкой на 8 тыс. изображений. Использовались те же настройки обучения и вывода, что и в COCO, за исключением того, что модели обучались в течение 300 эпох.

IV. Сравнение

Сравним две нейросети, OpenPose и Karao, используя набор данных COCO и рассмотрим несколько метрик, включая среднюю точность (AP) по различным пороговым значениям сходства ключевых точек объектов (OKS).

Используем нейросети на изображениях, подобранных авторами (Рис.8).

А

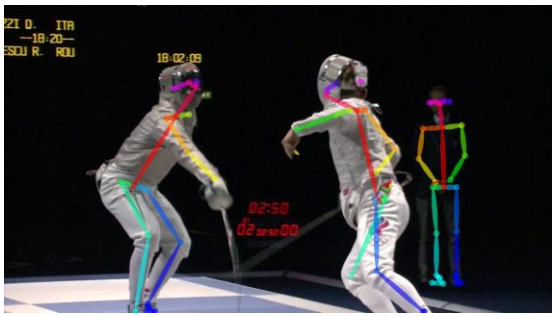
Б

В

Г

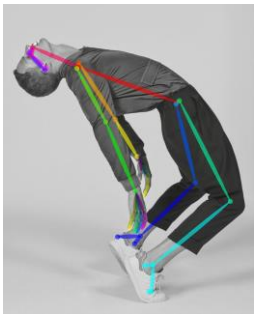
Рис. 8. Примеры изображений, используемых для сравнения

Полученные изображения после использования OpenPose и Karao представлены на Рис.9 и Рис.10.



А

Б

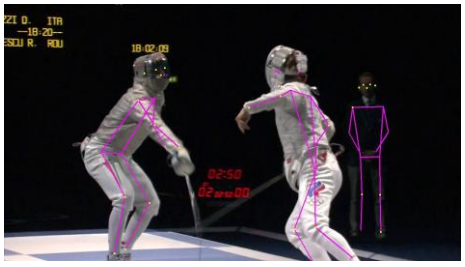


В

Г

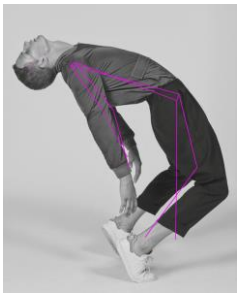
Рис. 9. Примеры изображений, полученные после обработки нейросетью OpenPose

OpenPose справилась с определением положения тела на Рис.9 (А). На Рис.9 (Б) возникла ошибка с определением ноги, но были определенны кисти. На Рис.9 (В) также отобразила кисти и стопы. Некорректно определила положения тела человека на Рис.9 (Г).



А

Б



В

Г

Рис. 10. Примеры изображений, полученные после обработки нейросетью Karao

Karao справилась с определением положения тела на Рис.9 (А, Б). На Рис.9 (В) неточно определила положение конечностей и некорректно отобразила одну из ног. Некорректно определила положения тела человека на Рис.9 (Г).

Таблица 1 отражает метрики для оценки OpenPose и Karao, полученные с использованием набора данных COCO.

ТАБЛИЦА 1. Метрики OpenPose и Karao

Модель	Задержка (мс)	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
OpenPose	74.1	63.8	85.9	70.3	62.0	69.1
Karao	67.0	63.0	86.3	69.5	58.0	70.8

В первую очередь, рассмотрим характеристику задержки функционирования обеих моделей. Из данных представленной в Таблице 1 следует, что модель Karao проявляет меньшую задержку в работе (67.0 мс), по сравнению с моделью OpenPose (74.1 мс). Это свидетельствует о возможности Karao обрабатывать изображения более эффективно с точки зрения временных затрат, что имеет значение в контексте операций в реальном времени или в условиях ограниченных вычислительных ресурсов.

Используя оценку сходства ключевых точек (OKS), при рассмотрении различных пороговых значений IoU, OpenPose демонстрирует более высокую среднюю точность (AP) при более высоких значениях IoU (0.75) и IoU 0.50:0.95, в то время как Karao превосходит OpenPose при более низких значениях IoU (0.50).

Кроме того, при анализе масштабов можно заметить, что OpenPose имеет более высокую среднюю точность (AP) при средних масштабах объектов (AP^{M}), в то время как Karao показывает лучшие результаты при более высоких масштабах (AP^{L}).

V. Заключение

В данной статье были рассмотрены наборы данных COCO и MPII, на которых осуществлялось обучение и последующее тестирование нейронных сетей. Были приведены две нейронные сети OpenPose и Karao, каждая из указанных сетей была рассмотрена в контексте своей архитектуры, а также используемых для обучения и тестирования наборов данных.

Представленные нейросети были подвергнуты тестированию и сравнению на наборе данных COCO. В рамках сравнения был проведен анализ, используя метрики, временных задержек и оценки сходства ключевых точек (OKS).

По полученным результатам нейросети справляются с задачей определения положения тела человека и имеют схожие значения средней точности. Karao имеет большее быстродействие, но OpenPose имеет возможность определять кисти и ступни человека на изображениях. В следствии эффективность нейросетей зависит от требований и задач.

Литература

[1] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[2] Tanchenko, A.P., Fedulin, A.M., Bikmaev, R.R. et al. UAV Navigation System Autonomous Correction Algorithm Based on Road and River Network Recognition. Gyroscopy Navig. 11, 293–299 (2020). https://doi.org/10.1134/S2075108720040100

[3] R. R. Bikmaev, A. A. Polukarov and R. N. Sadekov, "Visual Localization of a Ground Vehicle Using a Monocamera and Geodesic-Bound 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] Толстенко Л.С., Клейменов А.А., Али Б., Крынецкая Г.С., Коробков А.А. Анализ нейронных сетей для детектирования светофоров на изображениях // известия института инженерной физики. — 2023. — № 2(68). — С. 59-65.

[5] 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.

[6] Артем Евгеньевич Павликов, Михаил Геннадиевич Городничев, Обзор технологий определения положения тела человека //

Модели, системы, сети в экономике, технике, природе и обществе. 2023. №3 (47)

[7] Vítor Albiero, Xingyu Chen, Xi Yin, Guan Pang, Tal Hassner, "img2pose: Face Alignment and Detection via 6DoF, Face Pose Estimation,"

[8] Embodied Scene-aware Human Pose Estimation Zhengyi Luo, Shun Iwase, Ye Yuan, Kris Kitani, 2022

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In ICCV, 2017

[10] Zuhe Li, Mengze Xue, Yuhao Cui, Boyi Liu, Ruochong Fu, Haoran Chen, Fujiao Ju Lightweight 2D Human Pose Estimation Based on Joint ChannelCoordinate Attention Mechanism, 2023

[11] Fast QuadTree-Based Pose Estimation for Security Applications Using Face Biometrics: 12th International Conference, NSS 2018, Hong Kong, China, August 27-29, 2018, Proceedings Paola Barra, Carmen Bisogni, Michele Nappi, Stefano Ricciardi

[12] Linear Algorithms for Object PoseEstimation T. N. Tan, G. D. Sullivan and K. D. BakerIntelligent Systems GroupDepartment of Computer ScienceUniversity of Reading

[13] Comparison of Markerless and Marker-Based MotionCapture Technologies through Simultaneous DataCollection during Gait: Proof of ConceptElena Ceseracciu., Zimi Sawacha., Claudio Cobelli*Department of Information Engineering, University of Padova, Padova

[14] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, Winter 1989.

[15] 2D Human Pose Estimation: New Benchmark and State of the Art Analysis.Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler and Bernt Schiele. IEEE CVPR'14

[16] Microsoft COCO: Common Objects in Context, Tsung-Yi, Lin Michael, Maire Serge, Belongie Lubomir, Bourdev Ross, Girshick James, Hays Pietro, Perona Deva, Ramanan C. Lawrence Zitnick, Piotr Dollar

[17] Human Pose Estimation for Real-World Crowded Scenarios, Thomas Golda, Tobias Kalb, Arne Schumann, Jürgen Beyerer, IEEE, 2019

[18] OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, Yaser Sheikh, 2019

[19] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” in CVPR, 2017

[20] Rethinking Keypoint Representations: Modeling Keypoints and Poses as Objects for Multi-Person Human Pose Estimation, William McNally, Kanav Vats, Alexander Wong, John McPhee, 2022

[21] Learning non-maximum suppression, Jan Hosang, Rodrigo Benenson, Bernt Schiele, Max Planck Institut für Informatik, Saarbrücken, Germany

ГРНТИ 28.23.15
ASJC 1707
OECD 1.02.EP

Исследование возможности определения возраста клиента при помощи компьютерного зрения

И. И. Антипов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2306246@edu.misis.ru

***Аннотация* — в настоящее время вопрос автоматического распознавания возраста клиентов становится все более актуальным в современном обществе, а эффективные методы классификации клиентов по возрастным группам могут помочь в оптимизации всего процесса работы с ними. Цель данной статьи - провести исследование качества работы нейронных сетей для решения задач определения возраста. Для достижения поставленной задачи в работе рассматривается найденная в свободном доступе свёрточная нейронная сеть с открытым исходным кодом, основанная на библиотеке Keras и модели ResNet50, и возможность её применения на реальном наборе данных. В процессе проведения исследования сравнивается возможность определения возраста по изображениям из заготовленногодля обучения нейросети набора данных. Полученный после этого результат предобученной нейронной модели оценивается на отложенном (специально собранном) датасете.**

Ключевые слова — Компьютерное зрение, Глубокое обучение, Определение возраста, Распознавание лица, Keras, ResNet50

I. Введение

За последний год автоматическое распознавание возраста клиентов и покупателей стало трендом среди крупных иностранных магазинов, а также в настоящее время тестируется крупнейшими российскими ретейлерами. Одним из методов работы в данном направлении является определение принадлежности покупателя к одной из возрастных групп, и последующее разрешение или отказ на приобретение определённых товаров, либо создание персональных предложений и скидок, основанных на статистических опросах. Из самых известных ретейлеров, кто начал следовать данной тенденции,можно выделить такие европейские магазины как Bestway Retail, Innovative Technology, Bilka. Из российских представителей - X5, «Дикси» и «ВкусВилл» [1].

При создании классификатора возраста важной задачей является распознавание физических признаков возрастных отличий, относящих их по созданной автором модели классификации к одной из нескольких возможных возрастных групп, и совершение дальнейших действий в зависимости от поставленной задачи при работе с клиентами [2]. Для решения данной задачи применяются технологии компьютерного зрения [3, 4].

Определение возрастной группы включает в себя непосредственно распознавание характерных физических свойств рассматриваемого объекта, в данном случае клиента – черты лица, морщины, разрез глаз, цвет волос и прочие отличительные детали, характерные только для определенной возрастной группы [5]. В литературе приводится несколько способов распознания данных признаков, в том числе с использованием инфракрасных и лазерных сенсоров [6].

Методы глубокого обучения показали высокую производительность и способность к обобщению в задачах данного типа – особенно таких как распознавание и классификация [7]. В связи с этим, существует множество любительских разработок, представляющих собой детекторы в различных областях – наземное дорожное движение [8], железнодорожный транспорт, летательные аппараты [9], медицина, биология, городская инфраструктура [10], научная деятельность [11] и множество других сфер [12]. Один из подобных детекторов (с открытым исходным кодом), созданный специально для определения возраста людей по предоставленным изображениям, в данной работе будет рассматриваться и анализироваться с целью определения его пригодности к работе в условиях, приближенных к реальным.

Подходы, основанные на обучении, особенно те, которые используют глубокое обучение, требуют больших объёмов аннотированных данных [13]. В настоящее время в свободном доступе есть большое количество зарубежных наборов данных с классифицированными по возрасту изображениями людей. В таких датасетах используются фотографии как популярных людей, так и самых обыкновенных. Однако следует также заметить, что подходы, основанные на обучении, требуют больших вычислительных мощностей и времени [3, 4].

II. Наборы данных

Для обучения и тестирования рассматриваемой в данной работе нейросети использовался уже как готовый набор данных из открытого источника, так и локальный, собранный вручную для объективности результатов проведения тестирования. Рассмотрим используемые наборы

A. ChaLearn Looking at People

Обширный набор данных для обучения нейросети

определению возраста клиентов был взят с сайта ChaLearn Looking at People. Данный сборник содержит изображения известных и обычных людей из разных эпох, проиндексированных с указанием их реального возраста. В общей сложности данный набор хранит 7590 фотографий различных людей.

Рис. 1. Примеры подготовленных изображений людей из различных возрастных групп

B. Набор данных для контроля точности

Второй набор данных используется непосредственно для проверки рассматриваемой нейросети, полученной в результате обучения на первом наборе данных, к реальной работе. Второй набор не используется в самом процессе обучения, но на нём проводится дальнейший контроль точности полученной нейронной модели. В этом наборе данных изображения собраны с различных источников – фотографии из интернета, других наборов данных, фотографии людей, сделанные вручную автором статьи.Подобная выборка позволяет оценить, насколько обученная модель приспособлена к работе с реальными людьми, а не только с теми, на которых обучалась [5]. В данном наборе хранится 1710 фотографий.

Рис. 2. Примеры изображений людей из второго набора данных для контроля точности работы нейронной сети

III. НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА

Свёрточная нейронная сеть в Keras

Для достижения поставленной задачи в данном решении используется свёрточная нейронная сеть ResNet50, поставляемая в открытой библиотеке Keras. Сверточная нейронная сеть (на английском языке Convolutional Neural Network или CNN), и, в частности, ResNet50, является распространенной архитектурой глубокого обучения, которая обрабатывает данные с применением операции свертки, когда как Keras предоставляет простой и интуитивно понятный программный интерфейс для создания и обучения сверточных нейронных сетей [3]. Сам процесс обучения и тестирования состоит из 4 этапов.

Первый этап - выполняется загрузка и предобработка данных для обучения модели нейронной сети для определения возраста по изображениям. Загружаются метки из файла CSV с разделителем ";" (то есть, предполагается, что файл имеет формат CSV с разделителем ";") и пропускаются начальные пробелы при чтении данных. Создается объект, который будет выполнять аугментацию изображений в процессе обучения. Это помогает увеличить разнообразие данных для обучения и улучшить обобщающую способность модели. Затем создается генератор данных, который будет генерировать пакеты данных для обучения нейронной сети. Он использует данные, которые мы загрузили на первом шаге. Размер изображений изменяется до 224x224 пикселей, и обрабатывается пакетами по 32 изображения. Классовый режим установлен как 'raw', что означает, что метки не изменяются. Доля данных для обучения установлена в 75%. Далее функция возвращает генератор данных, который может быть использован для обучения модели.

На этом этапе происходит загрузка изображений и их соответствующих меток, подготовка их в виде данных, пригодных для обучения модели нейронной сети.

Второй этап – выполняется загрузка и предобработка данных для тестирования модели нейронной сети после завершения обучения. Выполняются аналогичные действия для предобработки данных, но с некоторыми отличиями. Как и в предыдущем фрагменте, здесь загружаются метки из CSV файла, используя разделитель ";" и пропуская начальные пробелы. Создается объект для аугментации данных тестового набора. Опять же, это помогает

нормализовать значения пикселей изображений и улучшить обобщающую способность модели. После создается генератор данных для тестового набора. Он также использует данные из каталога с изображениями и параметры для предварительной обработки. Однако в этом случае используется подмножество данных для валидации, установленное в 25%. Функция возвращает генератор данных, который может быть использован для оценки модели на тестовом наборе данных.

Таким образом, на втором этапе выполняется подготовка данных тестового набора для последующей оценки модели на изображениях лиц.

Третий этап - определяется функция create_model, которая и создает саму модель нейронной сети для определения возраста клиента. Процесс состоит из следующих шагов: сначала создается основная часть модели с использованием предварительно обученной нейросети ResNet50. Определяется форма входных изображений, загружаются предварительно обученные веса ResNet50, и исключается верхний слой из модели, поскольку пользователем будут добавляться свои слои. Создается последовательная модель Keras. Затем добавляется основная часть модели ResNet50 в модель Keras. Добавляется слой глобального пулинга. Этот слой усредняет признаки, полученные из предыдущего слоя, по всему пространству признаков, что позволяет значительно снизить количество параметров и предотвратить переобучение. Добавляется полносвязный слой с одним нейроном и функцией активации ReLU. Этот слой будет выводить предсказание возраста. Затем создается оптимизатор Adam с заданной скоростью обучения. Компилируется модель с оптимизатором Adam, функцией потерь среднеквадратичной ошибки (mean squared error) и метрикой средней абсолютной ошибки (mean absolute error). Функция возвращает созданную модель.

На данном этапе создается модель нейронной сети на основе архитектуры ResNet50 с добавлением нескольких слоев для решения задачи определения возраста по изображениям лиц.

Четвертый этап - определяется функция, которая используется для непосредственного обучения и последующего тестирования нейронной сети. Этот метод обучает модель на предоставленных данных. Создается генератор данных для обучения и генератор данных для валидации. Определяется размер пакета данных для обучения. Указывается количество эпох обучения. Определяется количество шагов, которые модель должна выполнить на каждой эпохе обучения и валидации соответственно. И также указывается, что вывод процесса обучения будет детализированным. После функция возвращает обученную модель.

Таким образом, данная функция используется для обучения модели нейронной сети на предоставленных данных и возвращает уже обученную модель, готовую для тестирования.

IV. Проведение испытаний

Для тестирования работоспособности выбранной нейронной модели сперва её требуется обучить. В данном случае переменная, настройка которой напрямую повлияет на успешность обучения это выбор количества эпох для тренировки нейросети. Эпоха обозначает один проход через все обучающие примеры в заданном наборе данных. Во время одной эпохи нейронная сеть проходит через все входные данные и обновляет веса своих параметров, чтобы минимизировать ошибку и улучшить свою производительность. Чем больше эпох, тем больше времени потребуется для обучения сети, но при этом повышается шанс достижения лучшей производительности [5, 6].

Для проведения обучения и тестирования было решено установить 15 эпох как оптимальное число, при котором время обучения модели будет длиться относительно быстро, но при этом итоговый результат будет показательным. Такое число эпох также выбрано в связи с ограниченной вычислительной мощностью оборудования испытателя. Автор данной нейронной модели при демонстрации работоспособности использует только 10 эпох, но подобное количество эпох является недостаточно показательным и не демонстрирует полную прогрессию и выход на стабильный уровень при обучении модели. Соотношение тестовых и валидационных данных составляет 3-к-1: 75% изображений используется для обучения, 25% для валидации.

Результат обучения в течение 15 эпох можно наблюдать на рисунках 3 и 4:

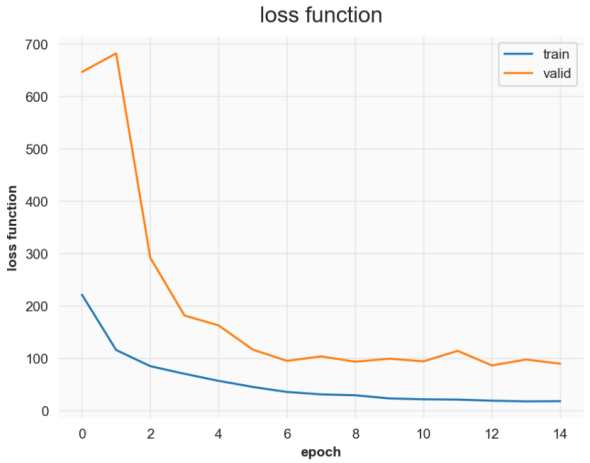


Рис. 3. Результат обучения и тестирования нейронной модели по показателю «Функция потерь»

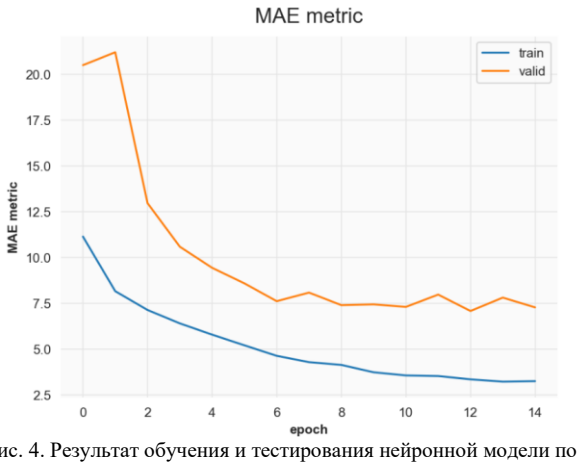


Рис. 4. Результат обучения и тестирования нейронной модели по показателю «Средняя абсолютная ошибка»

По графику функции потерь (рисунок 3) можно наблюдать, что на 6-й эпохе как в тестовых, так и в

валидационных данных расхождение данных вышло на относительно стабильный уровень с минимальным значением, незначительно изменяясь в течение последующих оставшихся эпох. В качестве функции потерь используется показатель «Среднеквадратичная ошибка», где чем показатель меньше, тем результат лучше.

По графику средней абсолютной ошибки (рисунок 4) видно, что на валидационных данных плато наблюдается уже на 12-й эпохе, таким образом можно сделать вывод, что 15 является оптимальным количеством эпох для качественного обучения модели, и в то же время данное число эпох позволяет избежания переобучения модели [5, 6] - явления, когда модель слишком точно подстраивается под тестовые данные, что приводит к плохой обобщающей способности на новых неизвестных раннее данных. Переобучение возникает, когда модель слишком сложна и слишком точно запоминает особенности обучающего набора данных, вместо обобщения свойств данных.

На валидационных данных выход на плато произошёл на 8-й эпохе с незначительными колебаниями. Параметр MAE рассчитывается как среднее абсолютных разностей между наблюдаемым и предсказанным значениями. Чем ближе показатель к нулю, тем точнее модель. MAE возвращается в том же масштабе значений, что и исходные данные, таким образом единицами измерения MAE выступает количество лет. По результатам ранее проведенных исследований в компаниях, наилучший показатель MAE составляет 1,5 года, а средний – 2,2 года [14]. На тестовых данных после 15-й эпохи данный показатель составляет 3.24 года, на валидационных – 7.27 лет.

Рис. 5. Распределение возраста в выборке

Наиболее вероятные причины данного поведения — либо начало переобучения, либо большее количество менее удачных валидационных данных из диапазона до 20 или после 40 лет (рисунок 5) по сравнению с данными, выпавшими во время обучения [5, 6]. В таком случае можно сделать два предположения для ещё большей оптимизации модели при необходимости. Первое это уменьшение количества эпох вплоть до 12, на которой наблюдалось наилучшее значение MAE. И второе предположение о необходимости добавления в тестовый набор данных большего количества фотографий людей младше 20 и старше 40 лет.

Для контроля точности работы полученной модели будет использоваться самостоятельно подготовленный набор данных, на котором не происходило обучения определителя возраста, но при этом второй набор содержит изображения, соответствующие заданной в нейронной модели возрастам. Таким образом, полученная нейросеть должна определять возраст людей на этих изображениях точно так же, как и во время обучения.

Второй набор данных состоит из 1710 фотографий, подавляющая их часть была сформирована из датасета, взятого с сайта huggingface. Данный датасет содержал 2500 фотографий людей с указанием их пола, возраста, и национальности. Были удалены размытые, нечеткие и затемненные изображения людей. Таким образом, осталось 1600 фотографий. Из разметки была удалена вся информация, кроме номера фотографии и возраста. Оставшиеся 110 фотографий были добавлены автором статьи – за образцы были взяты фотографии знаменитых личностей, как российских, так и иностранных. Таким образом, соотношение изображений по возрастам следующее: 130 фотографий в возрастном диапазоне от 1 до 10 лет, 180 от 10 до 20 лет, 410 от 20 до 30 лет, 490 от 30 до 40 лет, 320 от 40 до 50 лет, и 180 людей старше 50 лет.

Успешным результатом в данном случае будет считаться параметр MAE равный или чуть ниже 7.27 лет на валидационных данных, поскольку раннее было выяснено, что возможно имеет место недостаток тестовых данных и явление переобучения, что может вызвать проблемы с распознаванием новых изображений.

На рисунке 6 можно увидеть, что функция потерь дольше выходила на стабильный уровень, периодически совершая резкие скачки, но по итогу оказалась на уровне с тестовыми данными на рисунке 3. Результат на рисунке 7 демонстрирует, что с реальным набором данных нейронная модель вела себя более нестабильно, не имея четко выраженной стабильности. Тем не менее, итоговый результат показал точность метрики MAE 7.54 года, что в рамках проведения данного эксперимента с выявленными раннее недостатками модели и тестового набора данных является удовлетворительным результатом.

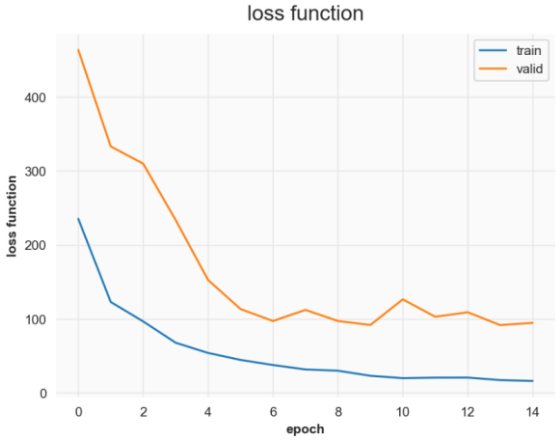


Рис. 6. Результат тестирования нейронной модели на реальных данных по показателю «Функция потерь»

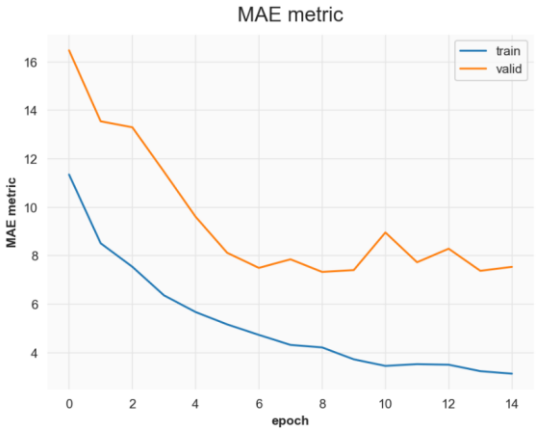


Рис. 7. Результат тестирования нейронной модели на реальных данных по показателю «Средняя абсолютная ошибка»

В таблице 1 приведены полученные значения метрик для 1-го и 2-го набора данных.

Таблица 1. Полученные метрики для наборов данных

	1 датасет	2 датасет
loss	15,39	16,53
mae	3,01	3,13
val_loss	93,97	95,05
val_mae	7,27	7,54

Основываясь на полученных результатах, можно сделать предварительный вывод о том, что обученный на первом наборе данных определитель возраста удовлетворительно справился с поставленной перед ним задачей и прошел контроль качества при помощи второго набора подготовленных данных, поскольку показатель MAE оказался на одном уровне [5]. Данный результат может быть улучшен корректировкой количества эпох обучения и добавлением новых данных в тестовую выборку [6].

V. Заключение

В рамках проведённого исследования было рассмотрено решение с открытым исходным кодом – определитель возраста клиента, основанный на принципе глубокого обучения с использованием библиотеки Keras и встроенной нейронной модели ResNet50 [7, 8], и также оценена его готовность к внедрению на реальных кассах самообслуживания в в крупных ретейлерах и других подобных местах, где такая технология могла бы быть применена.

Для тестирования пригодности определителя возраста к работе в реальных условиях использовались два набора данных – первый набор данных был заранее создан сообществом для свободного использования, второй набор данных был подготовлен ручную автором статьи для максимально объективной оценки полученных результатов [7]. Итоговые результаты исследования продемонстрировали, что предложенный в проекте алгоритм способен определять возраст со средней ошибкой в 7,5 лет.

Исходя из всего вышесказанного, можно подвести итог, что определитель возраста клиентов — это перспективная нейронная разработка, рабочие экземпляры которой уже активно создаются, внедряются и тестируются на рынке ретейлинга. Но рассмотренный в статье алгоритм решения нуждается в доработке путем модификации тестового набора данных новыми изображениями и корректировки количества эпох обучения, поскольку работающие алгоритмы имеют погрешность от 1,5 до 2,2 лет, что в сравнение с полученным при тестах показателем от 7,27 до 7,54 года демонстрирует значительную разницу в погрешности и на 241% является худшим показателем. После данных модификаций и получения приемлемых результатов с минимальной погрешностью, подобная система позволит автоматизировать процесс самообслуживания клиентов на кассах, без участия персонала магазина в нём.

Литература

[1] Ретейлеры X5, «Дикси» и «ВкусВилл» тестируют системы распознавания лиц, пола и возраста покупателей // Inc. URL: https://incrussia.ru/news/retejlery-testiruyut-sistemy-raspoznavaniya-lits/ (дата обращения: 22.042024).

[2] Аггарвал, Ч. Нейронные сети и глубокое обучение : учебный курс. – М. : Диалектика, 2020. – 744 с. : ил. – ISBN 978-5-907203-01-3.

[3] Джулли, Пал: Библиотека Keras - инструмент глубокого обучения / пер. с англ. А. А. Слинкин.- ДМК Пресс, 2017. – 249 с.

[4] Ян Эрим Солек. Программирование компьютерного зрения на языке Python / пер. с англ. Слинкн А. А. - М.: ДМК Пресс, 2016 - 312 с.: ил.

[5] 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.

[6] Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.. : Пер. с англ. - М. : Издательский дом ‘‘Вильямс’’, 2007. - 1408 с.

[7] Макшанов, А.В. Технологии интеллектуального анализа данных: Учебное пособие / А.В. Макшанов, А.Е. Журавлев. - СПб.: Лань, 2018. - 212 с.

[8] 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.

[9] 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

[10] 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.

[11] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-3-93-105.

[12] Полевой, Дмитрий & Kulagin, Petr & Ingacheva, Anastasia & Soldatova, Zhanna & Chukalina, Marina & Nikolaev, Dmitriy & Arlazarov, Vladimir. (2023). From tomographic reconstruction to automatic text recognition: the next frontier task for the artificial intelligence. 51. 10.1117/12.2680132.

[13] В. Ш. Берикашвили, С. П.Оськин Статистическая обработка данных, планирование эксперимента и случайные процессы : учебное пособие для вузов - 2-е изд., испр. и доп. - М. : Юрайт, 2021. - 163 с.

[14] В Британии тестируют нейросеть, определяющую возраст покупателя при покупке алкоголя // LIFE URL: https://life.ru/p/1469574 (дата обращения: 22.04.2024).

ГРНТИ: 28.23.15
ASJC: 1707
OECD: 01.02.EP

Распознавание текстовых CAPTCHA с помощью нейронных сетей

И. А. Антонов
кафедра инженерной кибернетики НИТУ МИСИС
Москва, Россия
m1908142@edu.misis.ru

***Аннотация*— веб-сайты могут повысить свою безопасность и предотвратить вредоносные интернет-атаки, используя CAPTCHA для определения того, является ли конечный пользователь человеком или роботом. Текстовая CAPTCHA является наиболее распространенной; она легко распознается людьми и трудно – машинами или роботами. Однако благодаря значительному прогрессу в области глубокого обучения становится намного проще создавать модели сверточных нейронных сетей (CNN) и других архитектур, которые могут эффективно распознавать текстовые CAPTCHA. В данной работе рассмотрены две нейросетевые модели, решающие задачу распознавания текстовых CAPTCHA, а также проведен их сравнительный анализ и инференс на собственноручно сгенерированных изображениях.**

Ключевые слова — CAPTCHA, компьютерное зрение, сверточные нейронные сети, глубокое обучение, обработка изображений, CNN.

I. Введение

Термин CAPTCHA — это аббревиатура, обозначающая «полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей». Капчи были впервые предложены Луи фон Аном и др. в 2003 году [1] и с тех пор стали обычной функцией безопасности в коммерческих приложениях для предотвращения вредоносных компьютерных программ и ботов. Эти тесты созданы так, чтобы они были сложными для компьютеров, но простыми для решения людьми. CAPTCHA может появляться во многих формах [2], включая текстовую [3], графическую [4] и звуковую [5]. Исследователи, изучающие распознавание изображений, играют важную роль в выявлении слабых мест в системах CAPTCHA, поскольку их исследования помогают разработчикам избежать этих уязвимостей в новых CAPTCHA, повышая безопасность в Интернете.

Несмотря на то, что текстовые CAPTCHA являются популярным и эффективным методом защиты от вредоносных компьютерных программ, эффективность текстовых CAPTCHA можно повысить, используя различные методы, такие как введение фонового шума, манипулирование текстом путем деформации, вращения, изменения длины и комбинирования символов. Тем не менее, с развитием технологий глубокого обучения современные защитные системы утратили способность работать с системами распознавания CAPTCHA. Следовательно, крайне важно создать более сложные меры безопасности, чтобы сделать текстовые CAPTCHA более устойчивыми к злонамеренным атакам.

Глубокое обучения [6,7,8,9] во многом способствовало широкому распространению технологии распознавания CAPTCHA. Использование нейронных сетей показало высокую эффективность в выявлении важных особенностей входных изображений и имеет широкий спектр применений в различных областях, таких как навигация беспилотников [10], распознавание еды [11], детекция повреждений при сборе урожая корнеплодов [12], распознавание дорожных знаков [13], предсказание движения дорожного транспорта [14]. Это делает подходы глубокого обучения желательной альтернативой созданию надежных алгоритмов, способных атаковать текстовые CAPTCHA. Хотя традиционные методы цифровой обработки изображений используются во многих алгоритмах распознавания CAPTCHA, они по-прежнему имеют ограничения, такие как недостаточная способность извлечения признаков и восприимчивость к шуму во входных изображениях [15]. Поэтому такие подходы постепенно заменяются более совершенными методами глубокого обучения.

В большинстве случаев алгоритмы распознавания текстовых CAPTCHA делятся на две группы [15]: основанные на сегментации и те, которые работают без сегментации. Методы, основанные на сегментации, обычно состоят из двух основных этапов: первый — это сегментация, при котором изображение CAPTCHA разбивается на отдельные символы, и второй — распознавание символов, при котором эти изолированные символы идентифицируются с помощью модуля распознавания символов. Этап сегментации имеет решающее значение, поскольку он оказывает существенное влияние на общую точность и эффективность системы. Однако многие алгоритмы сегментации имеют ограничения, приводящие к снижению эффективности и результативности. В результате исследователи начали изучать алгоритмы без сегментации в качестве альтернативы для преодоления недостатков, связанных с этим процессом.

В настоящее время популярность в сфере распознавания CAPTCHA приобрели модели без сегментации [15]. Эти модели имеют возможность напрямую распознавать и классифицировать символы CAPTCHA, не сегментируя их на отдельные блоки. Более того, модели без сегментации регулярно демонстрируют впечатляющий уровень точности и эффективности. Напротив, алгоритмы, использующие глубокое обучение для распознавания CAPTCHA, полагаются на обширные наборы данных для

эффективного распознавания особенностей CAPTCHA. Кроме того, алгоритмы без сегментации часто требуют сложной архитектуры и значительных объемов памяти, особенно при работе с CAPTCHA, содержащими множество символов.

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

II. Подходы к решению задачи распознавания текста

В данной работе рассматривается задача распознавания текста на фото CAPTCHA, что можно отнести к сфере оптического распознавания символов. В данной секции рассмотрим основные подходы, используемые исследователями в данной области.

Оптическое распознавание символов (OCR) — это электронное или механическое преобразование изображений рукописного или печатного текста в текст, закодированный машиной, будь то из отсканированного документа, фотографии документа, фотографии сцены или из текста субтитров, наложенного на изображение. Обычно система OCR включает в себя два основных модуля [16]: модуль обнаружения текста и модуль распознавания текста. Целью обнаружения текста является локализация всех текстовых блоков внутри текстового изображения либо на уровне слова, либо на уровне текстовой строки. Задача обнаружения текста обычно рассматривается как задача обнаружения объектов, к которой можно применить традиционные модели обнаружения объектов, такие как YoLOv5 и DBNet. Между тем, распознавание текста направлено на понимание содержимого текстового изображения и расшифровку визуальных сигналов в токены естественного языка. Задача распознавания текста обычно формулируется как задача кодировщика-декодировщика, где существующие методы используют энкодер на основе CNN (сверточной нейронной сети) для понимания изображений и декодер на основе RNN (рекуррентной нейронной сети) для генерации текста. Далее мы сосредоточимся на задаче распознавания текста

Недавний прогресс в распознавании текста связан с использованием преимуществ трансформерной архитектуры [16]. Однако существующие методы по-прежнему основаны на CNN, где блок самовнимания (self-attention) надстраивается над CNN в качестве кодировщиков для понимания текстового изображения. Для декодеров обычно используется нейросетевая темпоральная классификация (Connectionist Temporal Classification, CTC) в сочетании с внешней языковой моделью на уровне символов для повышения общей точности. Несмотря на большой успех, достигнутый гибридным методом кодирования/декодирования, все еще есть много возможностей для улучшения с помощью предварительно обученных моделей компьютерного зрения и обработки естественного языка: во-первых, параметры сети в существующих методах обучаются с нуля с использованием синтетических/размеченных человеком наборов данных, в связи с чем большие предобученные модели остаются неиспользованными и неисследованными для данных задач; во-вторых, поскольку трансформеры становятся все более и более популярными, появляется возможность и необходимость исследовать, могут ли предобученные трансформерные модели заменить CNN-часть современных методов.

Относительно простой задачей является распознавание текста в документах, так как там слова написаны единым шрифтом на едином фоне. Значительно сложнее распознавать номерные или дорожные знаки на улице или текстовые CAPTCHA, поскольку нужно учитывать дополнительные факторы [17]:

- Расположение: текст может находиться в случайном месте изображения, к тому же он может быть повернут и искажен.
- Шум: в текстовых CAPTCHA присутствует много артефактов.
- Шрифты: могут использоваться разные шрифты, в том числе, текст может быть рукописным.
- Блочность текста: на страницах книг или документов весь текст организован в блоки из линий, что нельзя гарантировать для уличных фото или CAPTCHA.
- Алфавит: фото могут содержать участки текста на разных языках, а в текстовых CAPTCHA помимо букв используются и цифры.

Практически со всеми этими факторами сталкиваются исследователи, занимающиеся распознаванием текста на CAPTCHA, поэтому далее приведем инструменты, использованные в данной работе для решения поставленной задачи.

III. Использованные нейронные сети

В данной работе для решения задачи распознавания текстовых CAPTCHA были использованы нейронные сети TrOCR от Microsoft [16] и Keras OCR от создателей Keras [18, 19].

A. TrOCR

TrOCR – это трансформерная модель OCR на основе для распознавания текста с предварительно обученными моделями CV и NLP (рисунок 1) [16]. В отличие от существующих моделей распознавания текста, TrOCR является простой, но эффективной моделью распознавания текста, не использующей CNN в качестве основы. Вместо этого она сначала изменяет размер входного текстового изображения до 384×384, а затем изображение разбивается на последовательность патчей размера 16×16, которые используются в качестве входных данных для трансформеров. Стандартная трансформерная архитектура с механизмом самовнимания используется как на энкодере, так и на декодере, где части слова генерируются как текст, распознанный из входного изображения. Для обучения модели TrOCR энкодер можно инициализировать с помощью предобученных моделей, таких как ViT, а декодер можно инициализировать предобученными моделями, такими как BERT. Таким образом, TrOCR имеет тройное преимущество. Во-первых, TrOCR использует предобученные модели трансформеров изображений и текста, в связи с чем нет необходимости во внешних языковых моделях. Во-вторых, TrOCR не требует какой-либо сверточной сети в качестве основы и не вносит никаких индуктивных смещений в изображения, что делает модель очень простой в реализации и обслуживании. В-третьих,

результаты экспериментов с тестовыми наборами данных OCR показывают, что TrOCR может достигать state-of-the-art результатов на наборах данных печатных, рукописных и текстовых изображений без каких-либо сложных шагов предварительной или постобработки. Кроме того, можно расширить TrOCR для распознавания многоязычного текста, используя многоязычные предобученные модели в части декодера и расширяя словарь.

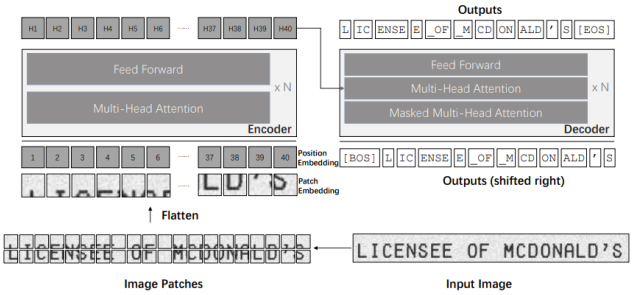


Рис. 1. Архитектура TrOCR, где модель энкодера-декодера разработана с использованием предобученного трансформера изображений в качестве энкодера и предобученного текстового трансформера в качестве декодера.

Рассмотрим прицип работы сети TrOCR более подробно.

1) Энкодер

Энкодер получает на вход изображение и приводит его к фиксированному размеру (H, W). Так как трансформерный энкодер не может работать с «сырыми» изображениями, не представляющими из себя последовательность входных токенов, энкодер преобразует входное изображение в батч из $N = \frac{HW}{P^2}$ квадратных патчей со стороной P. В свою очередь, гарантируется, что ширина W и высота H входного изображения (после приведения его к фиксированному размеру) делятся нацело на P. Впоследствии патчи преобразуются в векторы и линейно проецируются на D-мерные векторы (эмбеддинги патчей), где D — скрытый размер трансформера на всех его слоях.

В модели сохраняется специальный токен «[CLS]», который обычно используется для задач классификации изображений. Токен «[CLS]» объединяет всю информацию из всех эмбеддингов патчей и представляет все изображение. Также сохраняется токен дистилляции во входной последовательности при использовании предобученных моделей DeiT для инициализации энкодера.

В отличие от признаков, извлеченных с помощью CNN-подобной модели, трансформеры не имеют индуктивных смещений, специфичных для изображения, и обрабатывают изображение как последовательность патчей, что позволяет модели легче уделять различное внимание либо всему изображению, либо независимым патчам.

2) Декодер

В TrOCR использован оригинальный трансформерный декодер. В модуле внимания энкодера-декодера ключи и значения поступают с выхода энкодера, а запросы — с входа декодера. Кроме того, декодер использует маскировку внимания при самовнимании, чтобы не допустить получения во время обучения большего количества информации, чем при предсказании. Основываясь на том факте, что выходной сигнал декодера будет сдвигаться вправо на одну позицию от входа декодера, маска внимания должна гарантировать, что выход позиции i может принимать во внимание только предыдущий выходной сигнал, который является входными данными для позиций, идущих перед i:

$$h_i = Proj\big(Emb(Token_i)\big) \quad (1)$$

$$\sigma(h_{ij}) = \frac{e^{h_{ij}}}{\sum_{k=1}^{V} e^{h_{ik}}} \text{ для } j = 1,2,\dots,V \quad (2)$$

Скрытые состояния от декодера проецируются линейным слоем из измерения модели в измерение словаря размера V, а вероятности по словарю рассчитываются на основе функции softmax. Чтобы получить окончательный результат, используется лучевой поиск.

3) Инициализация моделей

Для инициализации энкодера в TrOCR используются модели DeiT и BEiT, трансформер обучается на ImageNET.

Для инициализации декодера в TrOCR используются модели RoBERTa и MiniLM. При загрузке этих моделей в декодеры структуры не совпадают точно, поскольку обе они являются трансформерными энкодерами архитектуры. Чтобы решить эту проблему, декодеры инициализировались с помощью моделей RoBERTa и MiniLM, вручную сопоставляя соответствующие параметры, а отсутствующие параметры задавались случайным образом.

B. Keras OCR

Keras OCR представляет собой простую модель, использующую функциональное API Keras. Она комбинирует архитектуры CNN и RNN и использует CTC loss. Архитектура такой модели представлена на рисунке 2.

Сеть Keras OCR состоит из следующих слоев:

1. Входной: принимает на вход изображения размера 200x50, содержащие 5-символьные текстовые CAPTCHA.
2. Сверточный: 32 фильтра, ядро 3x3, функция активации ReLU, функция инициализации ядра He Normal.
3. Max Pooling: ядро 2x2.
4. Сверточный: 64 фильтра, ядро 3x3, функция активации ReLU, функция инициализации ядра He Normal.
5. Max Pooling: ядро 2x2.
6. Reshape: изменение размера изображения.
7. Полносвязный: 64 нейрона, функция активации ReLU.
8. Dropout: доля 0,2.
9. Bidirectional: LSTM-слой, 128 ячеек, dropout 0,25.
10. Bidirectional: LSTM-слой, 64 ячейки, dropout 0,25.
11. Полносвязный: нейроны в количестве, равном объему словаря, функция активации SoftMax.
12. CTC: нейросетевая темпоральная классификация для декодирования результата работы сети.

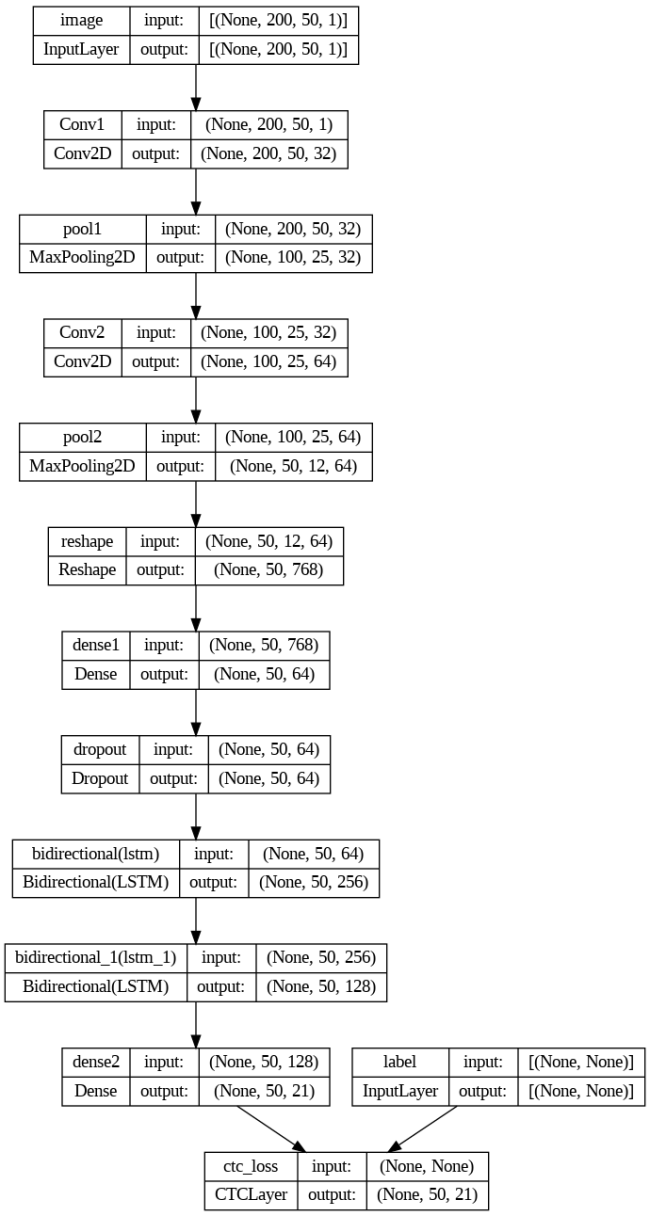


Рис. 2. Архитектура модели Keras OCR

IV. Набор данных и метрики

В данной работе был использован Captcha dataset [20].

Датасет содержит более 1000 текстовых CAPTCHA с подписями, пример одного такого файла представлен на рисунке 3.

Рис. 3. Пример текстовой CAPTCHA из использованного датасета

Для каждого такого файла требуется распознать последовательность символов и вывести ее.

Соответственно, в качестве функции оценки качества для данной задачи подходит CER (character error rate) [21]. Эта метрика является стандартной для задач распознавания речи (ASR) и оптического распознавания символов, она сравнивает тексты с позиции количества замен, удалений и вставок отдельных символов и рассчитывается как:

$$\mathbf{CER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \quad (3)$$

где S – количество замен символов,

D – количество удалений символов,

I – количество вставок символов,

N – общее количество символов в тексте,

C – количество правильных (совпавших) символов.

Данная метрика принимает значения от 0 до 1, где 0 означает, что сравниваемые тексты полностью совпали (модель распознавания текста отработала идеально), а 1 – тексты полностью отличаются (модель распознавания текста нуждается в существенных доработках).

Второй метрикой для данной задачи была выбрана *accuracy*, которую в данной задаче будем рассчитывать по формуле:

$$\mathbf{Accuracy} = \frac{c}{T} \quad (4)$$

где c – количество правильно распознанных CAPTCHA,

T – общее количество CAPTCHA в наборе.

V. Проведенные исследования и результаты

Перед использованием нейронных сетей выбранный датасет был разделен на тренировочную и тестовую выборки. Модель TrOCR была дообучена [22] на тренировочных изображениях, а модель Keras OCR – обучена с нуля. После этого была проведена валидация этих моделей на тестовой выборке и измерены показатели *accuracy* и CER.

Дообучение модели TrOCR происходило с использованием метода оптимизации Adam (а именно его реализации из [23]) на протяжении 57 эпох.

Обучение модели Keras OCR проходило на протяжении 60 эпох.

В таблице 1 представлены результаты работы выбранных нейронных сетей.

ТАБЛИЦА I. Полученные результаты

Модель	***Accuracy***	**CER**
TrOCR	0,990	0,006
Keras OCR	0,990	0,002

Мы видим, что обе нейронных сети справляются с залачей хорошо, но Keras OCR немного лучше.

После обучения и валидации моделей были сгенерированы собственные текстовые CAPTCHA с помощью библиотеки captcha [24] для языка Python (рисунки 4, 5, 6). Заметим, что они получились совершенно иного вида, чем в использованном датасете, а последняя CAPTCHA имеет больше символов.

Рис. 4. Сгенерированная CAPTCHA qw34r.png

Рис. 5. Сгенерированная CAPTCHA w1er3.png

Рис. 6. Сгенерированная CAPTCHA zx5p7m6.png

Результаты работы моделей на этих CAPTCHA представлены в таблицах 2 и 3.

ТАБЛИЦА II. Результаты работы моделей на сгенерированных образцах CAPTCHA

Изображение	**Предсказание TrOCR**	**Предсказание Keras OCR**
qw34r.png	wwgdf	f7[UNK][UNK][UNK]
w1er3.png	w2p7g	f7[UNK][UNK][UNK]
zx5p7m6.png	bx47nb	f7[UNK][UNK][UNK]

ТАБЛИЦА III. Значения метрик при работе моделей на сгенерированных образцах CAPTCHA

Модель	***Accuracy***	**CER**
TrOCR	0,000	0,765
Keras OCR	0,000	1,000

Мы видим, что обе модели справляются с данной задачей плохо. Тем не менее, модель TrOCR не «теряется», получая на вход непривычные данные, она смогла распознать некоторые символы, вероятно, за счет того, что она уже была предобучена перед использованием в задаче распознавания текстовых CAPTCHA. А модель Keras OCR, обученная только на одном датасете текстовых CAPTCHA, не смогла распознать текст на изображениях другого типа, выдавая одну и ту же ошибку на каждое из них.

VI. Заключение

В процессе работы была поставлена задача распознавания текстовых CAPTCHA, рассмотрены различные подходы к ее решению, подобран датасет и определены способы оценки качества.

Были использованы современные нейронные сети от известных разработчиков, одна из них была дообучена для более высокой производительности.

Обе нейронных сети прекрасно справились с распознаванием текстовых CAPTCHA во время тестирования их на изображениях исходного датасета, показав значение *accuracy* 0,990 и значения CER 0,006 и 0,002 для TrOCR и Keras OCR соответственно. Тем не менее, обе модели отработали плохо при получении на вход текстовых CAPTCHA вида, отличного от тех, которые присутствовали в датасете, - для обеих моделей *accuracy* составила 0 (поскольку ни одна CAPTCHA не была распознана правильно), а CER составила 0,765 для TrOCR и 1 для Keras OCR.

В дальнейшем планируется решить данную задачу, задействуя большее количество вычислительных мощностей. Это позволит использовать версию TrOCR с большим количеством параметров и/или более продвинутую архитектуру CNN-сети. Также это даст возможность обучать нейронные сети на гораздо большем датасете, который будет включать в себя различные виды и варианты текстовых CAPTCHA с разным количеством символов, в том числе и сгенерированные самостоятельно.

Литература

[1] Von Ahn L. et al. CAPTCHA: Using hard AI problems for security //Advances in Cryptology—EUROCRYPT 2003: International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4–8, 2003 Proceedings 22. – Springer Berlin Heidelberg, 2003. – С. 294-311.

[2] Kumar M., Jindal M. K., Kumar M. A systematic survey on CAPTCHA recognition: types, creation and breaking techniques //Archives of Computational Methods in Engineering. – 2022. – Т. 29. – №. 2. – С. 1107-1136.

[3] Singh V. P., Pal P. Survey of different types of CAPTCHA //International Journal of Computer Science and Information Technologies. – 2014. – Т. 5. – №. 2. – С. 2242-2245.

[4] Lupkowski P., Urbanski M. SemCAPTCHA—user-friendly alternative for OCR-based CAPTCHA systems //2008 international multiconference on computer science and information technology. – IEEE, 2008. – С. 325-329.

[5] Golle P., Ducheneaut N. Keeping bots out of online games //Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology. – 2005. – С. 262-265.

[6] Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks //Communications of the ACM. – 2017. – Т. 60. – №. 6. – С. 84-90.

[7] He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 770-778.

[8] Li J. et al. Tailings pond risk prediction using long short-term memory networks //IEEE Access. – 2019. – Т. 7. – С. 182527-182537.

[9] Zhou L. et al. Captcha recognition based on deep learning //Proceedings of the 4th International Conference on Big Data Research. – 2020. – С. 89-93.

[10] 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. – Т. 13. – №. 4. – С. 241-252.

[11] Kudryashov A. A., Mishchanin M. A., Sadekov R. N. Food recognition using deep learning networks and order history for smart canteen checkout automation.

[12] Osipov A. et al. Identification and classification of mechanical damage during continuous harvesting of root crops using computer vision methods //IEEE Access. – 2022. – Т. 10. – С. 28885-28894.

[13] Sadekov R. N. et al. Road sign detection and recognition in panoramic images to generate navigational maps //2017 24th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS). – IEEE, 2017. – С. 1-5.

[14] Guzhva N. S. et al. 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). – IEEE, 2022. – С. 1-6.

[15] Derea Z. et al. Deep Learning Based CAPTCHA Recognition Network with Grouping Strategy //Sensors. – 2023. – Т. 23. – №. 23. – С. 9487.

[16] Li M. et al. Trocr: Transformer-based optical character recognition with pre-trained models //Proceedings of the AAAI Conference on Artificial Intelligence. – 2023. – Т. 37. – №. 11. – С. 13094-13102.

[17] Изучение нейросетевого подхода к решению OCR на примере задачи распознавания арабского текста // Хабр URL: https://habr.com/ru/articles/682270/ (дата обращения: 24.04.2024).

[18] OCR model for reading Captchas // Keras: Deep Learning for humans URL: https://keras.io/examples/vision/captcha_ocr/ (дата обращения: 24.04.2024).

[19] yakhyo/captcha-reader-keras: OCR model for reading Captchas using Keras API // GitHub URL: https://github.com/yakhyo/captcha-reader-keras (дата обращения: 24.04.2024).

[20] (PDF) captcha dataset // ResearchGate URL: https://www.researchgate.net/publication/248380891_captcha_dataset (дата обращения: 01.04.2024).

[21] CER // HuggingFace URL: https://huggingface.co/spaces/evaluate-metric/cer (дата обращения: 24.04.2024).

[22] Transformers-Tutorials/TrOCR/Fine_tune_TrOCR_on_IAM_Handwriting_Database_using_native_PyTorch.ipynb at master · NielsRogge/Transformers-Tutorials // GitHub URL: https://github.com/NielsRogge/Transformers-Tutorials/blob/master/TrOCR/Fine_tune_TrOCR_on_IAM_Handwriting_Database_using_native_PyTorch.ipynb (дата обращения: 24.04.2024).

[23] Loshchilov I., Hutter F. Decoupled weight decay regularization //arXiv preprint arXiv:1711.05101. – 2017.

[24] captcha // PyPI URL: https://pypi.org/project/captcha/ (дата обращения: 24.04.2024)

Исследования методов распознавания текстовых документов с использованием компьютерного зрения

Д.В. Береснев
Кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2314671@edu.misis.ru

Аннотация — в данной статье рассматриваются современные методы распознавания текстовых документов с использованием технологий компьютерного зрения. Технология распознавания текстовых документов является полезной для автоматизации обработки большого объема информации, улучшения доступности данных и повышения эффективности работы в различных областях, таких как архивирование, цифровизация и поиск информации. В рамках исследования анализируются две нейросетевые модели, предназначенные для решения задач распознавания текстовых документов. Проводится сравнительный анализ их производительности и точности, а также проверка на собственном датасете, что позволяет выявить преимущества и недостатки каждой модели. Полученные результаты могут быть полезны для дальнейшего развития и оптимизации систем распознавания текстовых документов.

Ключевые слова — распознавание текстовых документов, компьютерное зрение, Tesseract, EasyOCR, нейросетевые модели, сравнительный анализ.

I. Введение

В последние годы нейронные сети и компьютерное зрение стали ключевыми технологиями, способствующими значительным достижениям в различных областях науки и техники. Нейронные сети, вдохновленные биологическими нейронами, представляют собой вычислительные модели, способные обучаться и делать предсказания на основе больших объемов данных [1-2]. Компьютерное зрение, в свою очередь, позволяет машинам интерпретировать и понимать визуальную информацию из окружающего мира, что открывает широкие возможности для автоматизации процессов, связанных с анализом изображений и видео [3].

Одной из важнейших задач в области компьютерного зрения является распознавание текста [4-5]. Эта технология позволяет автоматически извлекать текстовую информацию из изображений, что особенно актуально в современном мире, где большинство рабочих процессов в различных компаниях связаны с такими файлами, как счета, заказы, налоговые формы, финансовые отчеты, почта, анкеты и т. д.

Распознавание текстовых документов — это процесс преобразования изображений, содержащих текст, в редактируемый и поисковый текстовый формат [6]. Эта технология особенно полезна для автоматизации обработки документов, цифровизации архивов и улучшения доступности информации. С помощью распознавания текстовых документов можно значительно сократить время и усилия, необходимые для обработки большого объема данных [7].

Существует множество подходов к распознаванию текстовых документов, включая традиционные методы обработки изображений и современные нейросетевые модели. Традиционные методы часто основываются на правилах и алгоритмах, которые требуют ручной настройки и могут быть менее гибкими. В то же время нейросетевые модели, такие как Tesseract OCR и EasyOCR, предлагают более адаптивные и точные решения, обучаясь на больших объемах данных и улучшая свои результаты с течением времени.

В данной статье проводится обзор и сравнительный анализ двух нейросетевых моделей для распознавания текстовых документов: Tesseract OCR и EasyOCR. Основная цель исследования заключается в оценке производительности и точности этих моделей при решении задач распознавания текстовых документов.

II. Наборы данных

В рамках исследования будут использованы три датасета. Два из них — это открытые датасеты на английском языке, содержащие изображения с текстом и соответствующие текстовые данные для сравнения. Третий датасет — собственный, на русском языке, включающий фотографии документов. Эти датасеты позволят провести всесторонний анализ и оценить производительность моделей Tesseract OCR и EasyOCR как на английских, так и на русских текстовых документах.

A. ICDAR2015-SmartDoc

Датасет ICDAR2015-SmartDoc был создан в рамках соревнования ICDAR2015 competition on smartphone document capture and OCR (SmartDoc) - Challenge 2. Этот датасет предназначен для оценки производительности моделей распознавания текста на изображениях документов, сделанных с помощью смартфонов.

Датасет включает в себя 12.2 гигабайт данных и содержит 3630 фотографий документов. Каждое изображение сопровождается текстовым описанием, которое содержит точный текст, присутствующий на документе. На Рисунке 1 представлен пример данных датасета.

Рис. 1. Пример датасета ICDAR2015-SmartDoc

B. The IIIT 5K-word dataset

Датасет был собран с помощью поиска изображений в Google, используя запросы, такие как рекламные щиты, вывески, номера домов, таблички с названиями домов и постеры фильмов. В результате было собрано 5000 изображений, содержащих обрезанные слова из текстов на сценах и цифровых изображений. Датасет разделен на тренировочную и тестовую части, что позволяет использовать его для обучения и оценки моделей распознавания текста.

Этот датасет особенно полезен для задач распознавания обрезанных слов из большого лексикона. В дополнение к изображениям, датасет включает лексикон, содержащий более 0.5 миллиона слов из словаря, что делает его ценным ресурсом для разработки и тестирования моделей распознавания текста, таких как Tesseract OCR и EasyOCR. Пример изображений датасета представлен на Рисунке 2.

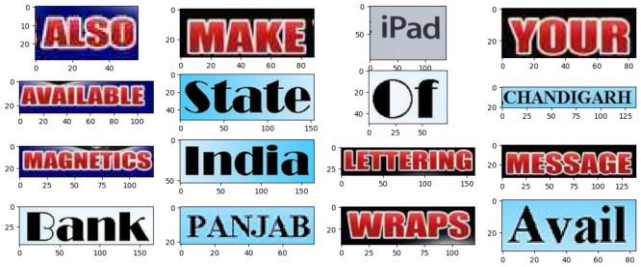


Рис. 2. Пример датасета The IIIT 5K-word dataset

C. Персональный набор данных

Персональный набор данных включает 12 документов, содержание которых было сгенерировано моделью GPT-4o. Текст этих документов написан на русском языке, использует грамматически верные конструкции, но не несет никакой смысловой нагрузки. Документы были сфотографированы в условиях плохого освещения, что добавляет дополнительную сложность для задач распознавания текста. Для каждого документа предоставлена транскрипция, что позволяет точно оценить производительность моделей Tesseract OCR и EasyOCR при работе с русскоязычными текстами в неблагоприятных условиях съемки. Пример данных представлен на Рисунке 3.

Рис. 3. Пример персонального датасета

III. Предобработка данных

Предобработка данных является важным этапом в процессе распознавания текста, так как качество входных данных напрямую влияет на точность и эффективность модели [8]. Ниже представлены основные этапы предобработки.

A. Сканирование и загрузка изображений

Первым шагом является получение изображений, которые будут использоваться для распознавания текста. Это могут быть отсканированные документы, фотографии страниц книг или снимки экранов. Важно обеспечить высокое качество изображений, чтобы минимизировать количество ошибок на последующих этапах.

B. Преобразование в оттенки серого

Цветные изображения часто содержат избыточную информацию, которая не нужна для распознавания текста. Поэтому изображения преобразуются в оттенки серого. Это упрощает дальнейшую обработку и уменьшает объем данных, с которыми работает модель.

C. Бинаризация

Бинаризация — это процесс преобразования изображения в черно-белый формат. Этот шаг помогает выделить текстовые элементы, отделяя их от фона. Наиболее распространенным методом бинаризации является пороговая обработка, при которой пиксели изображения сравниваются с определенным пороговым значением и преобразуются либо в черный, либо в белый цвет.

D. Удаление шума

Изображения могут содержать различные виды шума, такие как пятна, линии или артефакты, которые могут мешать распознаванию текста. Для удаления шума используются фильтры, такие как медианный фильтр или гауссово размытие. Эти методы помогают сгладить изображение и убрать мелкие дефекты.

E. *Выравнивание и коррекция наклона*

Документы могут быть отсканированы или сфотографированы под углом, что приводит к искажению текста. Для коррекции наклона используется алгоритм выравнивания, который определяет угол наклона текста и поворачивает изображение таким образом, чтобы текст стал горизонтальным. Это улучшает точность распознавания.

F. *Нормализация размера и разрешения*

Изображения могут иметь разные размеры и разрешения, что может затруднить работу модели. Нормализация включает изменение размера изображений до стандартного формата и приведение разрешения к единому значению. Это обеспечивает консистентность данных и улучшает производительность модели.

G. *Сегментация текста*

Сегментация текста — это процесс разделения изображения на отдельные текстовые блоки, строки и символы. Этот шаг важен для точного распознавания, так как модель должна обрабатывать текст по частям. Сегментация может быть выполнена с помощью различных методов, таких как проекционные профили или алгоритмы кластеризации.

Пример бинаризации и сегментации представлен на Рисунке 4.

Рис. 4. Пример бинаризации и сегментации

IV. Подходы к распознаванию текста

Распознавание текста (OCR, Optical Character Recognition) — это процесс преобразования изображений, содержащих текст, в машинно-читаемый текст. Существует несколько подходов к распознаванию текста, каждый из которых имеет свои особенности и области применения [9].

A. *Шаблонное распознование*

Шаблонное распознавание — один из самых ранних и простых методов распознавания текста, который основывается на сравнении фрагментов изображения с заранее подготовленными шаблонами символов [10].

Первым шагом является создание набора шаблонов для каждого символа, который нужно распознавать. Шаблоны могут быть созданы для различных шрифтов, размеров и стилей символов. Каждый шаблон представляет собой изображение символа в бинарной или градационной форме [11].

После предобработки и сегментации текста, полученные символы сравниваются с шаблонами. Сравнение может происходить как с помощью корреляции (формула 1),

$$r = \frac{\sum(Ai-\overline{A})(Bi-\overline{B})}{\sqrt{\sum(Ai-\overline{A})^2\sum(Bi-\overline{B})^2}})1($$

так и с помощью расчета расстояния (формулы 2 и 3).

$$dEuclidean(A,B) = \sqrt{\sum i = 1^n(Ai-Bi)^2})2($$

$$dcosine(A,B) = 1 - \frac{A\cdot B}{\|A\|\|B\|})3($$

где A и B — это два вектора, между которыми вычисляется косинусное расстояние.

B. *Методы на основе признаков*

Методы распознавания текста на основе признаков используют различные характеристики и признаки изображений для идентификации и классификации символов. Эти методы могут быть более устойчивыми к искажениям и вариациям в шрифтах по сравнению с методами шаблонного распознавания [12].

Основные концепции распознавания включают в себя извлечения признаков из изображения, которые могут использоваться для классификации. Примеры признаков включают:

- Гистограммы направлений градиентов
- Моменты изображений
- Контуры и формы

После извлечения признаков происходит классификация символов с использованием нейронных сетей или алгоритмов машинного обучения, таких как метод опорных векторов или классификаторы на основе дерева решений.

C. *Методы на основе скрытых марковских моделей*

Скрытые марковские модели (HMM) используются для моделирования последовательностей символов и слов. Эти модели хорошо справляются с задачами, где важен контекст, например, при распознавании рукописного текста. HMM могут быть объединены с другими методами, такими как нейронные сети, для улучшения точности [13].

D. *Методы на основе графов*

Данные методы представляют изображение текста в виде графа, где узлы соответствуют пикселям или группам пикселей, а ребра — их связям. Графовые методы могут быть полезны для распознавания сложных структур, таких как математические выражения или диаграммы [14].

E. *Структурные и топологические методы*

Данные методы распознавания текста основываются на анализе топологических свойств символов, таких как их форма, контуры, замкнутость и взаимное расположение

структурных элементов. Эти методы часто не зависят от масштаба, позиции и ориентации символов.

Каждый символ проходит процесс скелетизации, в результате чего полученный контур описывается в виде последовательного набора особых точек и цепного кода, где особые точки — точки, соседи которых образуют не менее трех связанных областей. На Рисунке 5 представлен пример скелетизации символа.

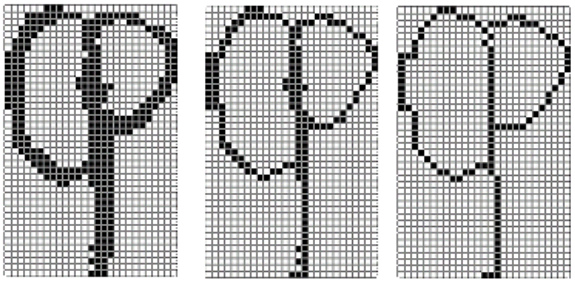


Рис. 5. Скелетизация образа символа

После скелетизации исходного текста производится огрубляющая предобработка, состоящая в удалении коротких линий, объединении близких триодов и уничтожении малых внутренних контуров. Далее с помощью перенумерации особых точек и изменения начала контура делается попытка классификации по одному из основных типов [15].

F. Методы на основе нейронных сетей

Методы на основе нейронных сетей включают конволюционные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и их комбинации. CNN используются для извлечения пространственных признаков из изображений текста и особенно эффективны для распознавания символов и слов благодаря своей способности обрабатывать двумерные данные. RNN, включая их модификации, такие как LSTM (Long Short-Term Memory), используются для обработки последовательностей данных и хорошо подходят для распознавания текста, так как могут учитывать контекст символов в строке. Комбинированные модели, такие как CRNN (Convolutional Recurrent Neural Networks), объединяют CNN и RNN для извлечения пространственных и временных признаков из изображений текста, что позволяет достигать высокой точности распознавания [16-19].

G. Методы на основе трансформеров

Трансформеры, такие как модели BERT и GPT, показали высокую эффективность в задачах обработки естественного языка (NLP). В OCR они могут быть использованы для постобработки распознанного текста, улучшая его качество за счет понимания контекста и исправления ошибок. Также модели, часто используемые в машинном переводе, могут быть адаптированы для OCR. Энкодер преобразует изображение в компактное представление, а декодер генерирует текст на основе этого представления. Такой подход позволяет эффективно обрабатывать длинные последовательности текста [20].

H. Гибридные методы

Гибридные методы комбинируют несколько подходов для достижения лучших результатов. Например, можно использовать CNN для извлечения признаков, HMM для моделирования последовательностей и трансформеры для постобработки текста. Гибридные методы часто показывают высокую точность и устойчивость к различным условиям съемки и типам текста.

V. Использованные нейронные сети

В ходе исследования использовались открытые и мультиязычные модели Tesseract и EasyOCR, реализующие гибридные методы распознавания текста.

A. Tesseract

Tesseract — это одна из самых мощных и популярных систем оптического распознавания символов (OCR), разработанная и поддерживаемая Google. Она является открытым программным обеспечение, поддерживает более ста языков и все основные форматы изображений. Tesseract также способен обрабатывать многостраничные документы и PDF-файлы [21].

Архитектура Tesseract представлена на Рисунке 6 и включает бинаризацию изображения, удаление шума и выравнивание текста, сегментацию и распознавания текста, а также постобработку.

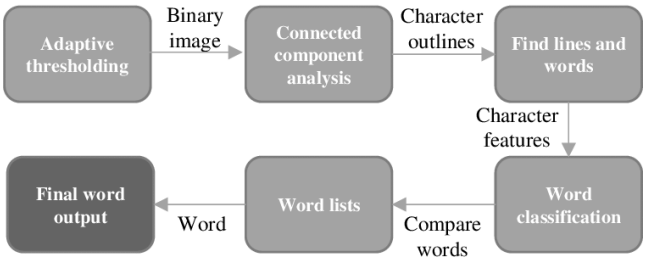


Рис. 6. Архитектура Tesseract

B. EasyOCR

EasyOCR — это современная библиотека для оптического распознавания символов (OCR), разработанная на базе PyTorch. EasyOCR является проектом с открытым исходным кодом и поддерживает более 80 языков. EasyOCR поддерживает основные форматы изображений и позволяет работать как с печатными, так и с рукописными текстами [22].

Архитектура библиотеки, представленная на Рисунке, включает предварительную обработку изображения, сегментацию, основанную на сторонних моделях, распознавания символов с помощью связки ResNet, LSTM и CTC моделей и постобработку, которая исправляет ошибки и группирует символы в слова и строки.

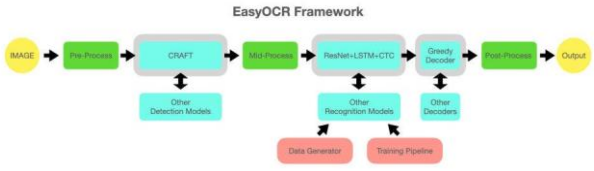


Рис. 7. Архитектура EasyOCR

VI. Метрики

Для сравнения моделей оптического распознавания символов (OCR) были использованы две основные метрики: Word Error Rate (WER) и Character Error Rate (CER). Эти метрики позволяют объективно оценить качество распознавания текста и сравнить производительность различных моделей.

A. WER

WER (Word Error Rate) — это метрика, используемая для оценки точности систем автоматического распознавания текста и речи. Она измеряет, насколько хорошо предсказанный машиной текст соответствует реальному тексту, и является одним из наиболее популярных методов оценки качества систем распознавания речи.

WER рассчитывается на основе трех типов ошибок (формула 4):

$$WER = \frac{S+D+I}{N})4($$

где S — количество замен (слово в предсказанном тексте заменено на другое),

D — количество пропусков (слово из исходного текста отсутствует в предсказанном тексте),

I — количество вставок (лишнее слово добавлено в предсказанный текст),

N — общее количество слов в исходном тексте.

Метрика WER является интуитивно понятной и широко применяется для систем распознавания, однако она не учитывает смысловые ошибки и является чувствительной к длине текста [23].

B. CER

Character Error Rate (CER) — это метрика, очень похожая на Word Error Rate (WER), но используемая для оценки качества распознавания текста на уровне символов (формула 5). Она особенно полезна в системах оптического распознавания символов, где точность распознавания отдельных символов имеет большое значение [24].

$$WER = \frac{S+D+I}{N})5($$

где S — количество замен (символ в предсказанном тексте заменен на другое),

D — количество пропусков (символ из исходного текста отсутствует в предсказанном тексте),

I — количество вставок (лишний символ добавлен в предсказанный текст),

N — общее количество символов в исходном тексте.

Данная метрика позволяет более точно оценить качество распознавания текста и обеспечивает более низкую чувствительность к длине текста, однако также не учитывает смысловые ошибки.

VII. Методы описательной статистики

Для сравнения моделей полученные данные необходимо интерпретировать, ниже представлены методы, которые использовались для оценки Tesseract и EasyOCR.

A. Mean (Среднее значение)

Среднее значение (или арифметическое среднее) — это сумма всех значений, деленная на количество этих значений (формула 6). Среднее значение дает общее представление о центре распределения данных.

$$\overline{X} = \frac{1}{N}\sum i = 1^{N} Xi)6($$

где $\overline{X}$ — среднее значение,

N — количество наблюдений,

Xi — значение каждого наблюдения.

B. Median (Медиана)

Медиана — это значение, которое делит отсортированный набор данных на две равные части, так что половина значений меньше медианы, а другая половина больше. Медиана менее чувствительна к выбросам по сравнению со средним.

C. Trimmed Mean (Обрезанное среднее)

Обрезанное среднее или усечённое среднее — это среднее значение после удаления заданного процента наибольших и наименьших значений из данных (формула 7). Этот метод помогает снизить влияние выбросов.

$$\overline{X}trim = \frac{1}{N-2k}\sum i = k + 1^{N-k} Xi)7($$

где N — количество наблюдений,

k — количество удаляемых значений с каждого конца,

X_i — значение каждого наблюдения после сортировки.

D. Standard Deviation (Стандартное отклонение)

Стандартное отклонение измеряет, насколько сильно значения разнятся от среднего значения (формула 8). Это ключевой показатель дисперсии или широты распределения данных.

$$\sigma = \sqrt{\frac{1}{N}\sum i = 1^{N}(Xi - \overline{X})^2})8($$

где σ — стандартное отклонение,

$\overline{X}$ — среднее значение,

N — количество наблюдений,

Xi — значение каждого наблюдения.

E. IQR (Межквартильный размах)

Межквартильный размах (IQR) — это разность между третьим (Q3) и первым (Q1) квартилями (формула 9). IQR используется для оценки разброса данных, устойчивого к выбросам, что делает его полезным для описания распределения данных.

$$IQR = Q3 - Q1)9($$

где Q1 — 25-й процентиль (первый квартиль),

Q3 — 75-й процентиль (третий квартиль).

VIII. Оценка точности

Результаты анализа моделей Tesseract и EasyOCR на основе 500 случайных фотографий из датасета ICDAR2015-SmartDoc представлены в таблице 1 для метрики WER и в таблице 2 для метрики CER. Для каждой

из метрик использованы основные методы описательной статистики.

Таблица 1 — Результаты анализа моделей Tesseract и EasyOCR на датасете ICDAR2015-SmartDoc (WER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.1501	0.1498	0.1503	0.0198	0.0334
Easy OCR	0.3800	0.3805	0.3792	0.0200	0.0329

Таблица 2 — Результаты анализа моделей Tesseract и EasyOCR на датасете ICDAR2015-SmartDoc (CER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.2102	0.2099	0.2105	0.0201	0.0332
Easy OCR	0.5703	0.5701	0.5707	0.0202	0.0333

Результаты работы моделей Tesseract и EasyOCR на основе 2000 случайных изображений из датасета The IIIT 5K-word dataset для метрики WER представлены в таблице 3, а для метрики CER — в таблице 4.

Таблица 3 — Результаты анализа моделей Tesseract и EasyOCR на датасете The IIIT 5K-word dataset (WER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.0102	0.0101	0.0100	0.0098	0.0134
Easy OCR	0.0898	0.0901	0.0900	0.0100	0.0150

Таблица 4 — Результаты анализа моделей Tesseract и EasyOCR на датасете The IIIT 5K-word dataset (CER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.0801	0.0803	0.0804	0.0101	0.0152
Easy OCR	0.1502	0.1501	0.1500	0.0099	0.0151

На базе персонального датасета, состоящего из 12 изображений текстовых документов, сфотографированных в условиях плохого освещения, были проведены тесты для определения эффективности моделей Tesseract и EasyOCR. Результаты по метрике WER (Word Error Rate) представлены в таблице 5. Результаты по метрике CER (Character Error Rate) представлены в таблице 6.

Таблица 5 — Результаты анализа моделей Tesseract и EasyOCR на персональном датасете (WER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.0005	0.0003	0.0005	0.0101	0.0168
Easy OCR	0.2212	0.2210	0.2212	0.0103	0.0148

Таблица 6 — Результаты анализа моделей Tesseract и EasyOCR на персональном датасете (CER)

	Mean	**Median**	**Trimmed**	**Std Dev**	**IQR**
Tesseract	0.0151	0.0150	0.0151	0.0104	0.0158
Easy OCR	0.3314	0.3310	0.3312	0.0099	0.0146

В таблице 7 представлены результаты средней производительности выполнения распознавания для моделей Tesseract и EasyOCR для одного изображения в каждом из рассмотренных датасетов.

Таблица 7 — Средняя производительность моделей Tesseract и EasyOCR для различных датасетов (время обработки одного изображения в секундах)

	ICDAR2015-SmartDoc	**The IIIT 5K-word**	**Персональный датасет**
Tesseract	45.68	0.45	4.12
Easy OCR	3.06	0.05	1.08

Из проведенного анализа видно, что модель Tesseract превосходит EasyOCR по всем рассмотренным метрикам, включая точность распознавания текста и стабильность результатов. Особенно заметно преимущество Tesseract по скорости работы, что делает её предпочтительным выбором для задач, требующих быстрой и точной обработки большого объема данных. Эти результаты подчеркивают эффективность модели Tesseract в различных сценариях распознавания текста.

IX. Заключение

В процессе работы были рассмотрены методы распознавания текстовых документов с использованием компьютерного зрения. Основное внимание уделялось анализу и сравнению двух популярных моделей для оптического распознавания текста — Tesseract и EasyOCR. Для проведения исследования использовались два открытых датасета: ICDAR2015-SmartDoc и The IIIT 5K-word dataset, а также был подготовлен персональный датасет из 12 изображений для финальной валидации моделей.

Основные метрики, использованные для оценки производительности моделей, включали Word Error Rate (WER) и Character Error Rate (CER). Эти метрики позволяют объективно оценить точность распознавания текста. Дополнительно были проанализированы такие статистические показатели, как среднее значение, медиана, усеченное среднее, стандартное отклонение и межквартильный размах (IQR). Также была проведена оценка средней производительности (скорости) выполнения распознавания для каждой модели на всех датасетах.

Результаты анализа показали, что модель Tesseract превосходит EasyOCR по всем рассмотренным метрикам, включая точность распознавания текста и стабильность результатов. Tesseract демонстрирует более низкие значения Mean WER и Mean CER по сравнению с EasyOCR на всех рассмотренных датасетах, что указывает на её более высокую точность распознавания текста. Кроме того,

меньшие значения стандартного отклонения и IQR для обеих метрик указывают на более стабильные и предсказуемые результаты модели Tesseract.

При оценке средней производительности выполнения распознавания, модель Tesseract также показала более высокие результаты, что делает её предпочтительным выбором для задач, требующих быстрой обработки большого объема данных.

Таким образом, результаты исследования показывают, что модель Tesseract является наиболее оптимальной для задач распознавания текста, обеспечивая как высокую точность, так и скорость обработки данных. Эти результаты подчеркивают эффективность модели Tesseract в различных сценариях распознавания текста и её преимущество перед EasyOCR.

Литература

1. A. A. Yakovlev, A. B. Kondybayeva and S. V. Solodov, "Intelligent System for Collecting, Analyzing and Managing Data in the Field of Medicine," *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, St. Petersburg, Russia, 2019, pp. 1-6, doi: 10.1109/WECONF.2019.8840588.
2. A. N. Semochkin, S. Zabihifar and A. R. Efimov, "Object Grasping and Manipulating According to User-Defined Method Using Key-Points," 2019 12th International Conference on Developments in eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 454-459, doi: 10.1109/DeSE.2019.00089.
3. 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.
4. D. V. Polevoy, P. A. Kulagin, A. S. Ingacheva, Zh. V. Soldatova, M. V. Chukalina, D. P. Nikolaev, V. V. Arlazarov, "From tomographic reconstruction to automatic text recognition: the next frontier task for the artificial intelligence," Proc. SPIE 12701, Fifteenth International Conference on Machine Vision (ICMV 2022), 127010P (7 June 2023); https://doi.org/10.1117/12.2680132
5. P. M. Manwatkar and S. H. Yadav, "Text recognition from images," 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-6, doi: 10.1109/ICIIECS.2015.7193210.
6. Pankaj Tripathi, "Top Text Recognition Algorithms: Enhancing OCR and IDP Capabilities", docsumo. Available at: https://www.docsumo.com/blog/text-recognition-algorithms
7. Aicha Fatrah, "How to Extract Information from documents: Template Matching", Medium. Available at: https://aicha-fatrah.medium.com/how-to-extract-information-from-documents-template-matching-e0540ae79599
8. Yasser Alginahi, "Preprocessing Techniques in Character Recognition", ResearchGate. Available at: https://www.researchgate.net/publication/221909023_Preprocessing_Techniques_in_Character_Recognition
9. Werawoolf, "Методы распознавания текста", Habr. Available at: https://habr.com/ru/articles/220077/
10. Roberto Brunelli, "Template Matching Techniques in Computer Vision", ResearchGate. Available at: https://www.researchgate.net/publication/252620698_Template_Matching_Techniques_in_Computer_Vision
11. Vijayarani Mohan, "Template Matching Technique for Searching Words in Document Images", ReseatchGate. https://www.researchgate.net/publication/329800652_Template_Matching_Technique_for_Searching_Words_in_Document_Images
12. Saif Safaa Shaker, Dhafer Alhajim, Ahmed Ali talib Al-khazaali, "Feature Extraction based Text Classification: A review", ResearchGate. Available at: https://www.researchgate.net/publication/361226607_Feature_Extraction_based_Text_Classification_A_review
13. Mounim A. El Yacoubi, "Hidden Markov Models for Text Recognition", ResearchGate. Available at: https://www.researchgate.net/publication/282505340_Hidden_Markov_Models_for_Text_Recognition
14. Edinei Peres Legaspe, Wellington Sperandio Silva, Caio Fernando Fontana, Eduardo Mario Dias, "Automatic character recognition based on graph theory: a new approach to automation", ResearchGate. Available at: https://www.researchgate.net/publication/262251470_Automatic_character_recognition_based_on_graph_theory_a_new_approach_to_automation
15. А.В. Афонасенко, А.И. Елизаров, "Обзор методов распознавания структурированных символов", CyberLeninka. Available at: https://cyberleninka.ru/article/n/obzor-metodov-raspoznavaniya-strukturirovannyh-simvolov
16. Afgani Fajar Rizky, Novanto Yudistira, Edy Santoso, "Text recognition on images using pre-trained CNN", arXiv: 2302.05105
17. Navdeep Singh Gill, "Convolutional Recurrent Neural Network For Text Recognition", xenonstack. Available at: https://www.xenonstack.com/insights/crnn-for-text-recognition
18. Rahul Agarwal, "Deep Learning Based OCR for Text in the Wild", nanonets. Available at: https://nanonets.com/blog/deep-learning-ocr/
19. Jasmin Kurtanovic, "Deep learning – OCR Text Detection and Text Recognition", Serengeti. Available at: https://serengetitech.com/tech/deep-learning-ocr-text-detection-and-text-recognition/
20. Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei, "Transformer-based Optical Character Recognition with Pre-trained Models", arXiv:2109.10282
21. R. Smith, "An Overview of the Tesseract OCR Engine," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.
22. M.A.M. Salehudin, S.N. Basah, H. Yazid, K.S. Basaruddin, M.J.A. Safar, M.H. Mat Som, K.A. Sidek, "Analysis of Optical Character Recognition using EasyOCR under Image Degradation", DOI: 10.1088/1742-6596/2641/1/012001
23. Youngja Park, Siddharth Patwardhan, Karthik Visweswariah, Stephen Gates, "An Empirical Analysis of Word Error Rate and Keyword Error Rate", ResearchGate. Available at: https://www.researchgate.net/publication/221488965_An_Empirical_Analysis_of_Word_Error_Rate_and_Keyword_Error_Rate
24. Utsav Poudel, Aayush Man Regmi, Zoran Stamenkovic, S.P. Raja. "Applicability of OCR Engines for Text Recognition in Vehicle Number Plates, Receipts and Handwriting", ResearchGate. Available at: https://www.researchgate.net/publication/376023699_Applicability_of_OCR_Engines_for_Text_Recognition_in_Vehicle_Number_Plates_Receipts_and_Handwriting

ГРНТИ 28.23.15
ASJC 1707
OECD 1.02.EP

Классификация земного покрова и землепользования

Д. И. Грищенко
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2309680@edu.misis.ru

***Аннотация*—данная статья посвящена актуальной проблеме классификации земного покрова и землепользования. Использование спутниковых снимков земной поверхности дает возможность оценивать характер использования природных ресурсов, плотность застройки. Методы глубокого обучения показали высокую производительность и точность в данной области. В работе рассматривается решение с открытым исходным кодом и оценивается точность распознавания способа использования земли на основе спутниковых изображений.**

Ключевые слова — компьютерное зрение, глубокое обучение, сверточная нейронная сеть, Transformer, распознавание спутниковых снимков, управление окружающей средой, Swin Transformer, EuroSAT.

I. Введение

Последнее время все острее встает вопрос сохранения экологии, влияния человека на природу. Вырубка лесов, загрязнение водоемов, бесконтрольная застройка, недобросовестное использование сельскохозяйственных земель — всё это угрозы, которые необходимо отслеживать и вовремя купировать.

Использование спутниковых изображений для классификации земного покрова и землепользования играет решающую роль и имеет важное значение в таких областях, как управление окружающей средой, землеустройство, развитие сельского хозяйства и сохранение природных ресурсов [1]. Однако пространственное распределение различных типов земного покрова является сложным, что делает повышение точности классификации сложной задачей.

Методы глубокого обучения показали высокую производительность и способность к обобщению во многих областях и типах задач, таких как классификация и обнаружение [2, 3, 4]. Сверточные нейронные сети хорошо себя показывают в подобных задачах, в том числе в распознавании изображений. В связи с этим, существует множество разработок, связанных с задачей распознавания, в таким областях как наземное дорожное движение [5], железнодорожный транспорт, летательные аппараты [6], медицина, биология, городская инфраструктура [7], научная деятельность [8] и в множестве других сфер. В работе рассматривается одно из подобных решений с открытым исходным кодом в области глубокого обучения для определения типа земной поверхности по спутниковым снимкам.

Подходы, основанные на обучении, особенно те, которые используют глубокое обучение, требуют больших объёмов аннотированных данных. В настоящей работе использовались готовые наборы данных, собранных с помощью спутниковой фотографии.

II. Наборы данных

Для обучения и тестирования рассматриваемой в данной работе нейронной сети использовались два набора данных.

A. EuroSAT

Набор данных представляет собой RGB-версию EuroSAT, которая основана на мультиспектральных изображениях со спутника Sentinel-2. Набор данных содержит в общей сложности 27 тысяч изображений и 10 категорий, каждая из которых содержит изображения с разрешением 64*64 пикселя, и каждый пиксель представляет пространственный охват в 10 метров [9].

Представлены следующие категории:

- AnnualCrop – однолетний урожай;
- Forest – лес;
- HerbaceousVegatation – травянистая растительность;
- Highway – шоссе;
- Industrial – промышленная зона;
- Pasture – пастбище;
- PermanentCrop – постоянный урожай;
- Residential – жилая застройка;
- River – река;
- SeaLake – озеро, море.

На рисунке 1 представлены примеры изображений, находящихся в наборе данных EuroSAT.

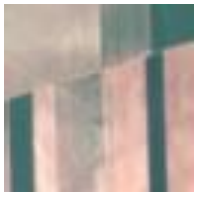
а  б 

Рис. 1. Примеры изображений в наборе данных EuroSAT: а), б) AnnualCrop, в) Forest, г), д) HerbaceousVegetation, е), ж) Highway, з) Industrial, и) Pasture, к) PermanentCrop, л) Residential,м) River, н) SeaLake

B. UC Merced Dataset

Набор данных содержит спутниковые снимки, распределенные по 21 классу, с разрешением 256 x 256 пикселей. Оригинальный датасет UC Merced Dataset содержит по 100 изображений на класс. За счёт аугментации число изображений в каждом классе было повышено до 500 [10].

Представлены следующие категории:

- Agricultural – сельское хозяйство;
- Airplane – самолёт;
- Baseballdiamond – бейсбольная площадка;
- Beach – пляж;
- Buildings – здания;
- Chaparral – колючий кустарник;
- Denseresidential – плотная жилая застройка;
- Forest – лес;
- Freeway – автострада;
- Golfcourse – поле для гольфа;
- Harbor – порт;
- Intersection – перекресток/пересечение дорог;
- Mediumresidential – жилая застройка средней плотности;
- Mobilehomepark – трейлерный парк;
- Overpass – эстакада;
- Parkinglot – парковка;
- River – река;
- Runway — взлётная полоса;
- Sparseresidential – редкая застройка;
- Storagetanks – резервуары для хранения;
- Tenniscourt — теннисный корт.

На рисунке 2 представлены примеры изображений, находящихся в наборе данных UC Merced Dataset.

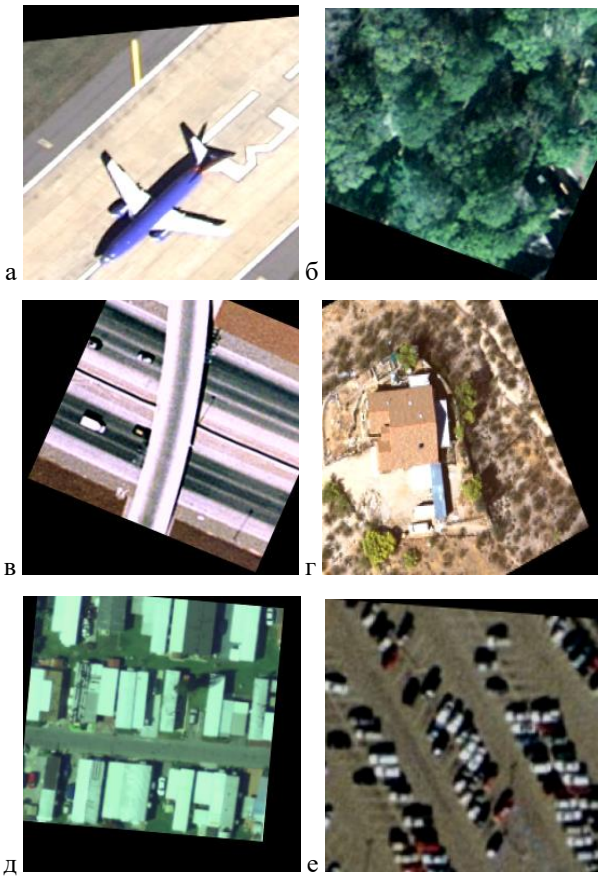


Рис. 2. Примеры изображений в наборе данных UC Merced Dataset: а) Airplane, б) Forest, в) Overpass, г) Spaseresidential, д) Mobilehomepark, е) Parkinglot

III. Нейросетевые архитектуры

Swin Transformer

Благодаря постоянному развитию вычислительной мощности и глубокому обучению сети классификации изображений на основе CNN (сверточных нейронных сетей) и трансформеров широко исследуются и применяются. Таким образом, эта работа направлена на использование методов глубокого обучения, в частности, путем интеграции сильных сторон как CNN, так и трансформеров, для достижения точной классификации земного покрова и землепользования.

В данной работе используется модель Swin Transformer – это предварительно обученный мощный алгоритм классификации изображений [11]. Алгоритм модифицируется и настраивается с использованием набора данных спутниковых изображений. Изображение сначала разбивается на участки, затем сглаживается и добавляется позиционное кодирование, а затем подается в модель Transformer для обучения с применением собственного внимания (self-attention). Классификация изображений выводится с использованием полностью связанных слоев.

Swin Transformer - это модель глубокого обучения, основанная на механизме самоконтроля, разработанная специально для обработки изображений [11]. По сравнению с традиционными сверточными нейронными сетями (CNNs) и Vision Transformer (ViT), Swin Transformer представляет новый оконный механизм, который разделяет изображение на серию перекрывающихся участков изображения и использует механизм самоконтроля для улавливания глобальных зависимостей между участками изображения. Этот оконный механизм поддерживает эффективность модели при обработке изображений большого размера, а также демонстрирует лучшую производительность при захвате зависимостей на больших расстояниях.

Полная архитектура рассматриваемой нейросети представлена на рисунке 3.

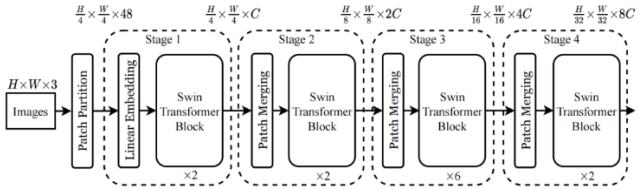


Рис. 3. Архитектура модели нейронной сети

Структура модели состоит из иерархических блоков Swin Transformer, каждый блок состоит из модуля самоконтроля с несколькими головками на основе смещенного окна (MSA), за которым следует 2-слойный MLP с нелинейностью GELU между ними. Слой LayerNorm применяется перед каждым модулем MSA и каждым MLP, а остаточное соединение применяется после каждого модуля [11].

По сравнению со сверточными нейронные сетями, Swin Transformer может фиксировать глобальные зависимости между участками изображения, не будучи ограниченным фиксированным размером поля восприятия, тем самым обладая более широкими возможностями моделирования. По сравнению с ViT, Swin Transformer обладает преимуществами в эффективности, улавливании зависимостей на больших расстояниях, гибкой обработке объектов и пригодности для обработки крупномасштабных данных изображений.

IV. Проведение тестирования

Для проведения тестирования необходимо настроить предобученную модель на используемый набор данных. Основным параметром при обучении является количество эпох. Эпоха обозначает один проход через все обучающие примеры в заданном наборе данных. Во время одной эпохи нейронная сеть проходит через все входные данные и обновляет веса своих параметров, чтобы минимизировать ошибку и улучшить свою производительность. Чем больше эпох, тем больше времени потребуется для обучения сети, но при этом повышается шанс достижения лучшей производительности [4, 12].

Для обучения модели на каждом из наборов данных было выбрано 20 эпох.

A. EuroSAT

График зависимости точности от эпох на обучающих данных EuroSAT представлен на рисунке 4.

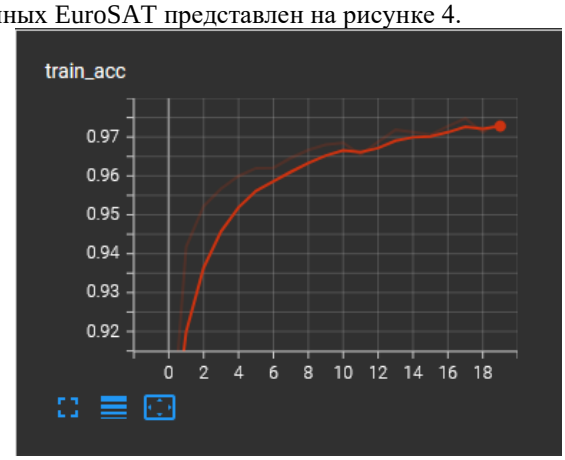


Рис. 4. График точности на обучающих данных EuroSAT

График зависимости точности от эпох на тестовых данных представлен на рисунке 5.

Рис. 5. График точности на тестовых данных EuroSAT

Лучшие показатели точности как на обучающих, так и на тестовых данных модель демонстрирует на 18 эпохе. Далее начинается спад показателей, что может говорить о переобучении модели.

Для дальнейшего тестирования была выбрана модель, полученная на 18 эпохе.

На рисунке 5 представлены примеры предсказания на проверочных данных.

Рис. 5. Примеры предсказания на проверочных данных EuroSAT

На всех четырех примерах предсказание модели совпало с меткой данных. Можно сделать вывод о том, что при модель обучена хорошо и при предсказании достигает высокой точности.

По результатам обучения на тестовых и проверочных наборах данных была построена матрица несоответствий. В строках матрицы расположены предсказанные моделью классы, в столбцах матрицы – установленные у данных метки.

На рисунке 6 представлена матрица несоответствий, полученная по результатам обучения модели на 18 эпохе.

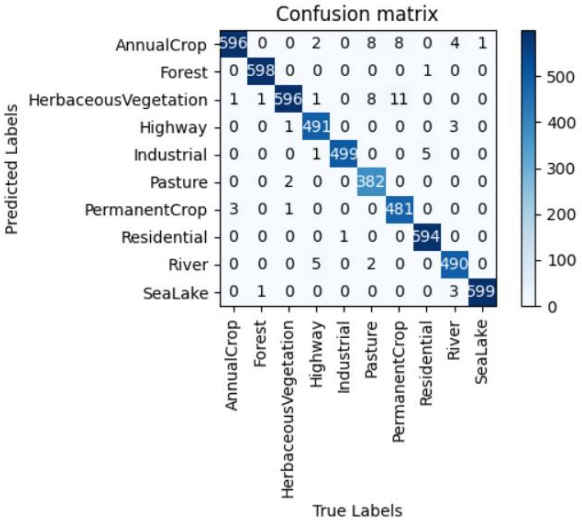


Рис. 6. Матрица несоответствий EuroSAT

Модель показывает очень высокую точность распознавания на тестовом наборе данных. Наибольший процент ошибок наблюдается в классе Pasture (пастбища) — 4.7%. Нейронная сеть в большинстве ошибочных случаев «путала» пастбища с однолетним урожаем (AnnualCrop) и травянистой растительностью (HerbaceousVegetation). В то же время наибольшая точность в предсказании была показана в классах Forest, Industrial и SeaLake – 0,33%, 0,2% и 0,17% соответственно. Данный результат можно связать со значительными визуальными отличиями цветовой палитры изображений в данных классах.

На рисунке 7 представлена таблица, содержащая параметры Precision (отношение положительных предсказаний к общему числу предсказаний), Recall (отношение положительных предсказаний к общему числу изображений данного класса) и Specificity (отношение положительного предсказания отсутствия класса к общему числу изображений, на которых класс отсутствует) по каждому классу.

```
the model accuracy is  0.9862962962962963
+----------------------+-----------+--------+-------------+
|                      | Precision | Recall | Specificity |
+----------------------+-----------+--------+-------------+
|      AnnualCrop      |   0.963   | 0.993  |    0.995    |
|        Forest        |   0.998   | 0.997  |     1.0     |
| HerbaceousVegetation |   0.964   | 0.993  |    0.995    |
|       Highway        |   0.992   | 0.982  |    0.999    |
|      Industrial      |   0.988   | 0.998  |    0.999    |
|       Pasture        |   0.995   | 0.955  |     1.0     |
|    PermanentCrop     |   0.992   | 0.962  |    0.999    |
|     Residential      |   0.998   |  0.99  |     1.0     |
|        River         |   0.986   |  0.98  |    0.999    |
|       SeaLake        |   0.993   | 0.998  |    0.999    |
+----------------------+-----------+--------+-------------+
```

Рис. 7. Таблица метрик

Соотношение данных метрик дает представление о высокой точности распознавания и низкой вероятности ошибок при совершении предсказаний на тестовом наборе данных.

Итоговая точность модели, полученной на 18 эпохе, составляет 0,9862.

B. UC Merced Dataset

График зависимости точности от эпох на обучающих данных UC Merced Dataset представлен на рисунке 8.

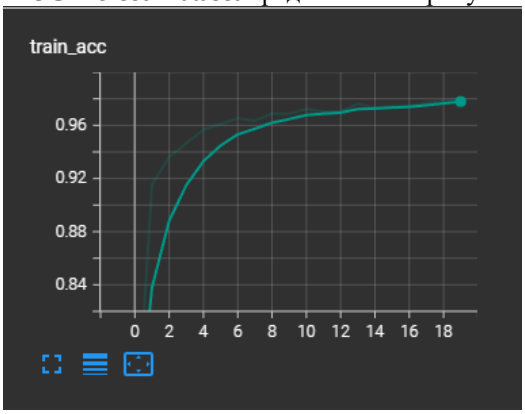


Рис. 8. График точности на обучающих данных UC Merced Dataset

График зависимости точности от эпох на тестовых данных UC Merced Dataset представлен на рисунке 9.

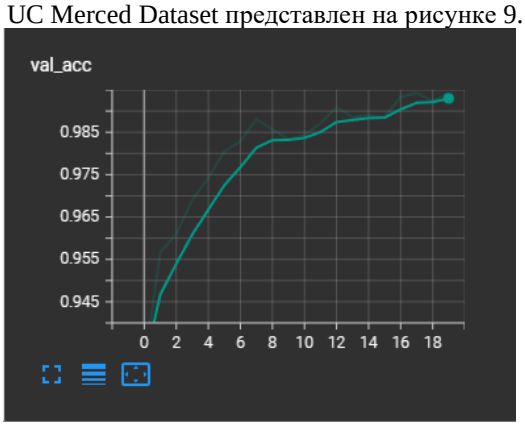
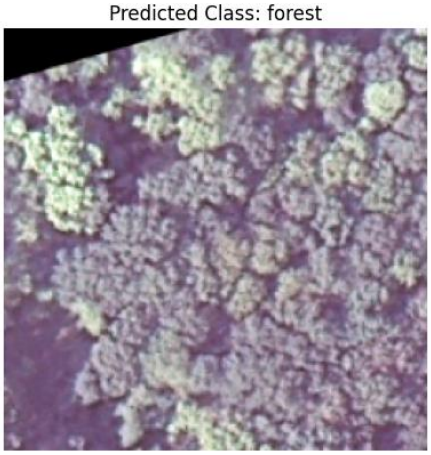


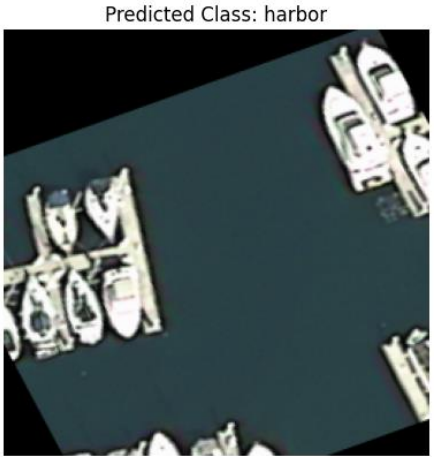
Рис. 9. График точности на тестовых данных UC Merced Dataset

Лучшие показатели точности как на обучающих, так и на тестовых данных модель демонстрирует на последней 20 эпохе.

Для дальнейшего тестирования была выбрана модель, полученная на 20 эпохе.

На рисунке 10 представлены примеры предсказания на проверочных данных.





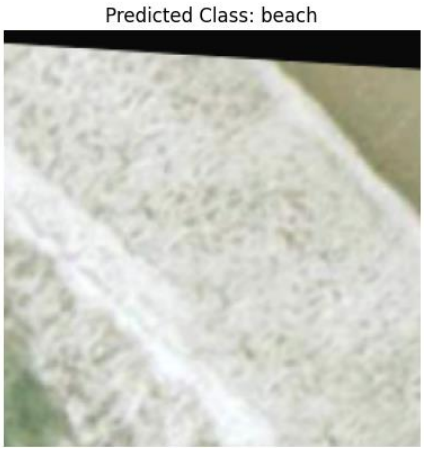


Рис. 10. Примеры предсказания на проверочных данных UC Merced Dataset

На всех четырех примерах предсказание модели совпало с меткой данных. Можно сделать вывод о том, что при модель обучена хорошо и при предсказании достигает высокой точности.

По результатам обучения на тестовых и проверочных наборах данных была построена матрица несоответствий. В строках матрицы расположены предсказанные моделью классы, в столбцах матрицы – установленные у данных метки.

На рисунке 11 представлена матрица несоответствий, полученная по результатам обучения модели на 20 эпохе.

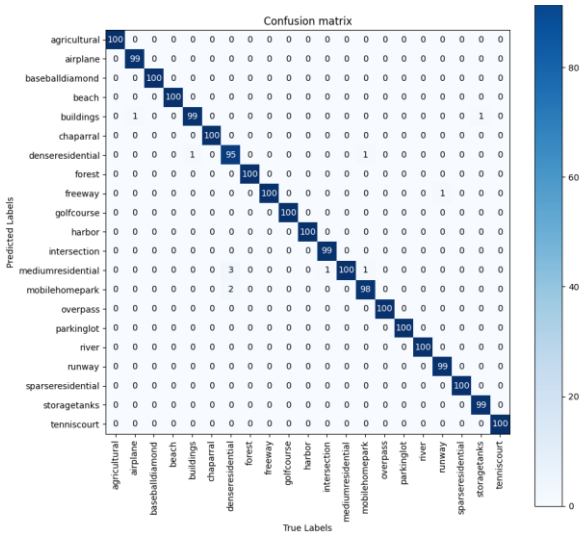


Рис. 11. Матрица несоответствий

При обучении на наборе данных UC Merced Dataset модель показывает крайне высокую точность при предсказании, вероятность ошибки по каждому классу не превышает 0,1. Такие показатели могут быть связано с значительными визуальными отличиями в цветах и формах объектов на изображениях разных классов.

На рисунке 12 представлена таблица, содержащая параметры Precision (отношение положительных предсказаний к общему числу предсказаний), Recall (отношение положительных предсказаний к общему числу изображений данного класса) и Specificity (отношение положительного предсказания отсутствия класса к общему числу изображений, на которых класс отсутствует) по каждому классу.

the model accuracy is 0.9942857142857143

	Precision	Recall	Specificity
agricultural	1.0	1.0	1.0
airplane	1.0	0.99	1.0
baseballdiamond	1.0	1.0	1.0
beach	1.0	1.0	1.0
buildings	0.98	0.99	0.999
chaparral	1.0	1.0	1.0
denseresidential	0.979	0.95	0.999
forest	1.0	1.0	1.0
freeway	0.99	1.0	1.0
golfcourse	1.0	1.0	1.0
harbor	1.0	1.0	1.0
intersection	1.0	0.99	1.0
mediumresidential	0.952	1.0	0.998
mobilehomepark	0.98	0.98	0.999
overpass	1.0	1.0	1.0
parkinglot	1.0	1.0	1.0
river	1.0	1.0	1.0
runway	1.0	0.99	1.0
sparseresidential	1.0	1.0	1.0
storagetanks	1.0	0.99	1.0
tenniscourt	1.0	1.0	1.0

Рис. 12. Таблица метрик

Соотношение данных метрик дает представление о практически 100% точности распознавания загородных объектов и низкой вероятности ошибки при распознавании зданий и городской застройки, не превышающей 5%.

Итоговая точность модели, полученной на 20 эпохе, составляет 0,99438.

V. Заключение

В рамках проведенного исследования было рассмотрено готовое решение с открытым исходным кодом по классификации земного покрова и землепользования на основе спутниковых фотографий поверхности. Нейронная сеть была построена на основе модели Swin Transformer.

Для обучения и тестирования были использованы набор данных EuroSAT, содержащий 27 тысяч изображений, разделенных на 10 категорий земной поверхности, и набор данных UC Merced Dataset, содержащий 10500 изображений, разделенных на 21 категорию.

По результатам сравнения работы модели на двух наборах данных, можно сделать вывод о том, что лучшие результаты были показаны при распознавании классов, имеющих визуальные особенности, например, лес, река, шоссе. И более плохие при распознавании похожих между собой объектов, например, модель может перепутать многолетние культуры и травянистую растительность, разные виды жилой застройки. При этом модель показывает отличную точность при классификации изображений, более 95%.

Полученные в результате обучения модели результаты показали крайне высокую эффективность при выполнении поставленной задачи. Обученная нейронная сеть может применяться для контроля использования земных ресурсов, разрастания городов, количества и типа выращиваемых культур, бесконтрольной вырубки лесов.

Литература

[1] Торсунова О.Ф. Использование данных космической съемки сверхвысокого разрешения для решения задач территориального зонирования // Вестник СГУГиТ. – 2018. – Т. 23 – № 2 – С. 219 – 230.

[2] Аггарвал, Ч. Нейронные сети и глубокое обучение : учебный курс. – М. : Диалектика, 2020. – 744 с. : ил. – ISBN 978-5-907203-01-3.

[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] 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.

[6] 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

[7] 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.

[8] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-3-93-105.

[9] EuroSat Dataset // Kaggle, available at: https://www.kaggle.com/datasets/apollo2506/eurosat-dataset (Accessed: May 25, 2024)

[10] Land-Use Scene Classification // Kaggle, available at: https://www.kaggle.com/datasets/apollo2506/landuse-scene-classification (Accessed: May 27, 2024)

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows // Microsoft Research Asia, 2021

[12] Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд.. : Пер. с англ. - М. : Издательский дом ‘‘Вильямс’’, 2007. - 1408 с

ГРНТИ коды
ASJC коды
OECD коды

Обнаружение кораблей на спутниковых изображениях с использованием компьютерного зрения

А. Д. Дедов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2313016@edu.misis.ru

Аннотация — в условиях растущей важности мониторинга морской деятельности и безопасности на воде, обнаружение кораблей на спутниковых изображениях становится все более актуальной задачей. В статье рассматривается метод обнаружения кораблей с использованием современных технологий компьютерного зрения. Основное внимание уделено применению алгоритма YOLO, который обеспечивает высокую точность и скорость распознавания объектов. Результаты экспериментов подтверждают высокую производительность и точность алгоритма YOLO в задаче обнаружения кораблей. Эти результаты подчеркивают потенциал применения данного метода в таких областях, как морская навигация, мониторинг окружающей среды и обеспечение безопасности на воде.

Ключевые слова — обнаружение судов, спутниковые изображения, компьютерное зрение, алгоритмы распознавания, морская навигация, мониторинг окружающей среды, безопасность на воде, YOLOv8

I. Введение

Обнаружение кораблей на спутниковых изображениях является важной задачей в области морской навигации, мониторинга окружающей среды и обеспечения безопасности на воде. С увеличением количества судов и ростом международных морских перевозок возрастает необходимость в эффективных и точных методах мониторинга и анализа морских пространств. Спутниковые изображения предоставляют обширные данные, которые могут быть использованы для отслеживания судов в реальном времени, выявления потенциальных угроз и предотвращения инцидентов.

Традиционные методы обнаружения кораблей, такие как визуальный анализ и использование радарных систем, имеют свои ограничения, включая высокую трудоемкость и зависимость от погодных условий. Современные технологии компьютерного зрения и глубокого обучения предлагают новые возможности для автоматизации и повышения точности этих процессов. Использование спутниковых изображений в сочетании с алгоритмами глубокого обучения позволяет значительно улучшить результаты обнаружения и идентификации объектов на воде [1].

В последние годы большое внимание уделяется развитию алгоритмов глубокого обучения, которые способны эффективно обрабатывать большие объемы данных и извлекать из них полезную информацию [2, 4]. Одним из таких алгоритмов является YOLO (You Only Look Once), который получил широкое признание благодаря своей способности быстро и точно распознавать объекты на изображениях [5, 6]. Этот алгоритм отличается высокой производительностью и может быть применен для различных задач [7], включая обнаружение кораблей на спутниковых снимках.

Настоящее исследование направлено на анализ эффективности алгоритма YOLO в задаче обнаружения кораблей на спутниковых изображениях. В рамках исследования проведен детальный обзор метода, а также экспериментальная оценка их результатов на реальных данных. Полученные результаты демонстрируют, что алгоритм YOLO способен значительно повысить точность и скорость распознавания кораблей, что открывает новые перспективы для применения в области морского мониторинга.

Таким образом, использование передовых технологий компьютерного зрения и глубокого обучения, таких как алгоритм YOLO, представляет собой значительный шаг вперед в решении задачи обнаружения кораблей на спутниковых изображениях. Дальнейшие исследования и разработки в этой области будут способствовать улучшению безопасности на воде и оптимизации процессов морской навигации.

II. Наборы данных

Для обучения и тестирования рассматриваемой в данной работе нейросети использовались некоторые наборы данных. Рассмотрим используемые открытые наборы.

A. OpenSARship-2.0

Набор данных состоит из 34,528 фрагментов изображений, которые были вырезаны из общего числа 87 изображений со спутника Sentinel-1. Эти изображения получены в режиме интерферометрической широкой полосы (IW). Датасет включает два доступных продукта режима IW: данные с одним взглядом (SLC) и данные с обнаружением наземного диапазона (GRD). Из этих 87 изображений Sentinel-1, 52 были получены из изображений GRD, а 35 - из изображений SLC, выбранных из 10 типичных сцен с интенсивным морским движением в различных частях мира за последние годы. Разрешение пространственной сетки изображений Sentinel-1 составляет от 2.7 × 22 до 3.6 × 22 метров и 20 × 22 метров. Изображения в датасете имеют смешанную поляризацию VV и VH. Каждое изображение корабля соответствует сообщению

автоматической идентификационной системы (AIS). Размеры изображений варьируются от 30 × 30 до 120 × 120 пикселей. Для каждого изображения Sentinel-1 SAR доступны четыре подкаталога, предоставляющие различные форматы фрагментов изображений: оригинальные данные, визуализированные данные в градациях серого, визуализированные данные в псевдоцветах и откалиброванные данные.

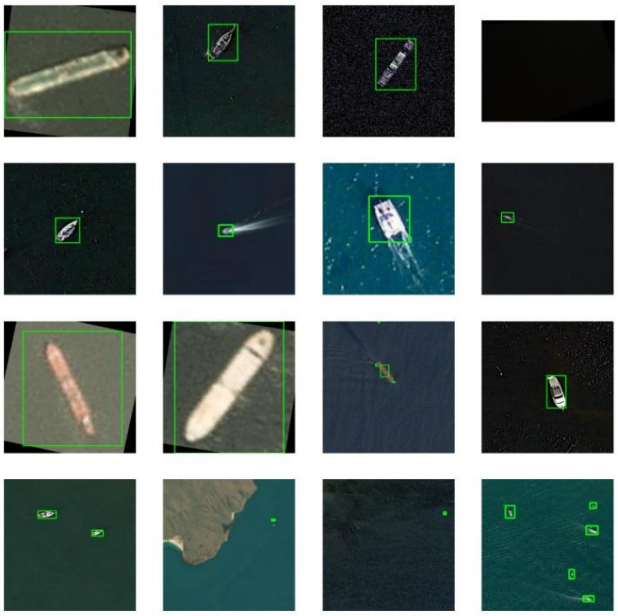


Рис. 1. Случайная выборка кадров из набора данных OpenSARship-2.0

B. Ships in Satellite Imagery

Является аналогом датасета *A*. Представляет собой набор изображений, извлеченных из спутниковых снимков, собранных над районами Сан-Франциско-Бей и Сан-Педро-Бей в Калифорнии. Включает в себя 4000 изображений размером 80x80 пикселей в формате RGB, помеченных как "ship" или " background ". Изображения происходят из полнокадровых визуальных сцен PlanetScope, орторектифицированных с размером пикселя 3 метра.

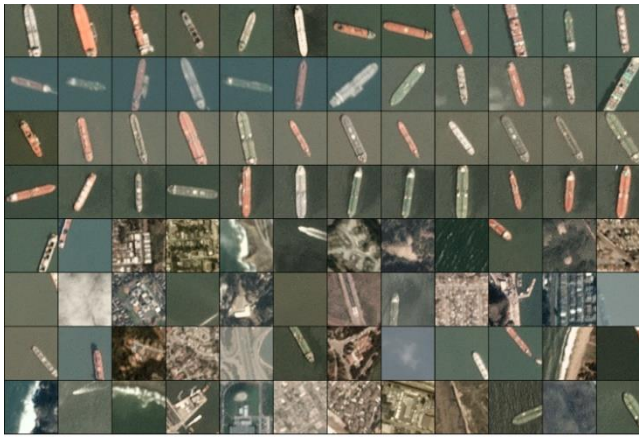


Рис. 2. Случайная выборка кадров из набора данных Ships in Satellite Imagery

III. Нейросетевая архитектура

Основное отличие алгоритма YOLO [7] от других методов на основе сверточных нейронных сетей (CNN) [9, 10], используемых для обнаружения объектов, заключается в его способности к быстрому распознаванию объектов в режиме реального времени. YOLO обрабатывает все изображение сразу, проходя через сверточную нейронную сеть только один раз, что и объясняет его название "You Only Look Once". В других алгоритмах изображение проходит через CNN многократно. Это дает YOLO преимущество в высокой скорости обнаружения объектов.

Проблема обнаружения объектов сложнее задачи классификации, которая может распознавать объекты, но не определяет их местоположение на изображении и не работает с изображениями, содержащими несколько объектов [3]. YOLO использует один прямой проход по сети для всего изображения, деля его на области и прогнозируя ограничивающие рамки и вероятности для каждой области. Эти рамки взвешиваются по предсказанным вероятностям. После применения non-max подавления (для обеспечения обнаружения каждого объекта только один раз) выводятся распознанные объекты с ограничивающими рамками.

YOLO позволяет одной CNN одновременно прогнозировать несколько ограничивающих рамок и вероятности классов для этих рамок [8, 11]. YOLO обучается на полных изображениях и напрямую оптимизирует производительность обнаружения, что дает модели ряд преимуществ перед другими методами.

В данной работе модель YOLOv8 была обучена для детекции кораблей. Нейронная сеть на выходе имеет два класса – «ship» и «background». Открытые наборы данных использовались для обучения, валидации и тестирования нейронной сети.

Архитектура YOLOv8 (рис. 5) представляет собой одну из версий популярной семейства алгоритмов для обнаружения объектов в реальном времени. Ниже представлен обзор основных черт архитектуры YOLOv8.

Основные принципы:

- Единоразовая обработка изображения: YOLOv8, как и предыдущие версии, обрабатывает всё изображение сразу, проходя через сверточную нейронную сеть только один раз. Это позволяет значительно ускорить процесс обнаружения объектов по сравнению с другими методами, которые обрабатывают изображение многократно.
- Одновременное предсказание рамок и классов: YOLOv8 одновременно прогнозирует несколько ограничивающих рамок и вероятности классов для этих рамок, используя один прямой проход по сети.

Архитектурные улучшения в YOLOv8

- Нейронная сеть CSPDarknet53 [12]: В YOLOv8 используется CSPDarknet53 в качестве базовой нейронной сети. Эта модифицированная версия Darknet включает блок "cross-stage" (CSP), что улучшает эффективность и обобщающую способность модели.
- Пирамидальная сеть (PANet [13]): Для объединения признаков на различных уровнях используется PANet, что помогает лучше обрабатывать объекты разных масштабов.

- SPP (Spatial Pyramid Pooling) [14]: В YOLOv8 применяется SPP для увеличения размера поля зрения, что улучшает обнаружение мелких объектов.

Конфигурации

- Разнообразие конфигураций: YOLOv8 предлагается в нескольких конфигурациях: YOLOv8-S, YOLOv8-M, YOLOv8-L и YOLOv8-XL. Каждая конфигурация имеет разные архитектурные особенности, такие как количество слоев и параметров, что позволяет выбирать модель в зависимости от требуемого уровня производительности и точности.

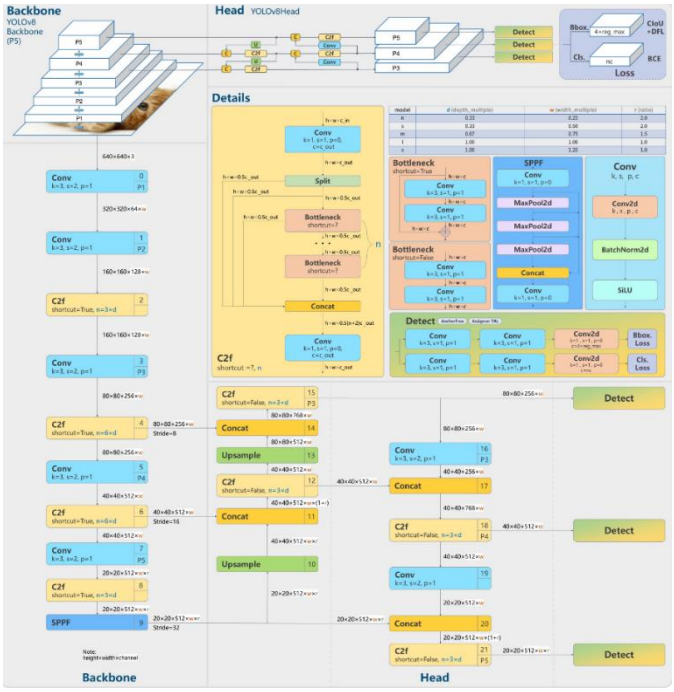


Рис. 3. Архитектура YOLOv8

Обучение

- Обучение на больших наборах данных: Модель обучается на обширных наборах данных, таких как COCO (Common Objects in Context) [15], что обеспечивает высокую обобщающую способность.
- Метод обучения: используется метод обучения с учителем (supervised learning) с применением размеченных данных, где модель учится определять классы и ограничивающие рамки объектов.

IV. Анализ результатов

Анализ результатов производительности детектирования кораблей включает в себя оценку точности, полноты обнаружения и других метрик для измерения эффективности модели [16].

Производительность модели:

- Train Box Loss: измеряет разницу между предсказанными ограничивающими рамками и фактическими рамками объектов в обучающих данных. Низкое значение указывает на более точное совпадение предсказанных рамок с реальными.
- Train Class Loss: измеряет разницу между предсказанными вероятностями классов и реальными метками классов объектов в обучающих данных. Низкое значение свидетельствует о более точном предсказании классов объектов.
- Train DFL Loss: измеряет расхождение между предсказанными и реальными картами признаков объектов в обучающих данных. Низкое значение указывает на более точное предсказание признаков объектов.
- Metrics Precision (B): измеряет долю истинных положительных срабатываний среди всех предсказанных ограничивающих рамок. Высокое значение означает, что модель лучше распознает истинные положительные срабатывания и минимизирует ложные положительные результаты.

$$Presicion = \frac{TP}{TP + FP}$$

- Metrics Recall (B): измеряет долю истинных положительных срабатываний среди всех фактических ограничивающих рамок. Высокое значение означает, что модель лучше распознает все истинные положительные объекты и минимизирует ложные отрицательные результаты.

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

- Metrics mAP50 (B): измеряет среднюю точность модели для различных категорий объектов при пороге пересечения IoU 50%. Высокое значение указывает на высокую точность и способность модели эффективно обнаруживать и локализовать объекты.
- Metrics mAP50-95 (B): измеряет среднюю точность модели для различных категорий объектов при порогах пересечения IoU от 50% до 95%. Высокое значение указывает на высокую точность модели при различных уровнях перекрытия между предсказанными и реальными объектами.

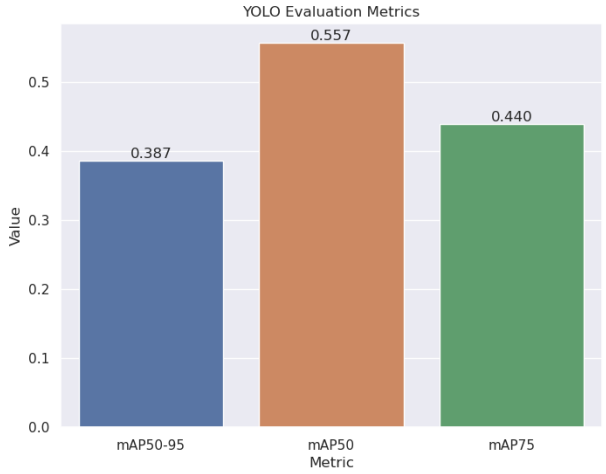


Рис. 4. Метрики лучшей модели

Mean Average Precision (mAP) [17], представленный на рисунке 4, является популярной метрикой оценки в задачах детектирования объектов, включая модель YOLO. Она применяется для определения точности модели

объектного детектирования через измерение её способности распознавать объекты на изображении, а также качества этих детекций. mAP учитывает как количество правильно обнаруженных объектов, так и точность распознавания, что делает её надежным показателем для оценки эффективности моделей обнаружения объектов.

Рис. 5. Метрики при обучении

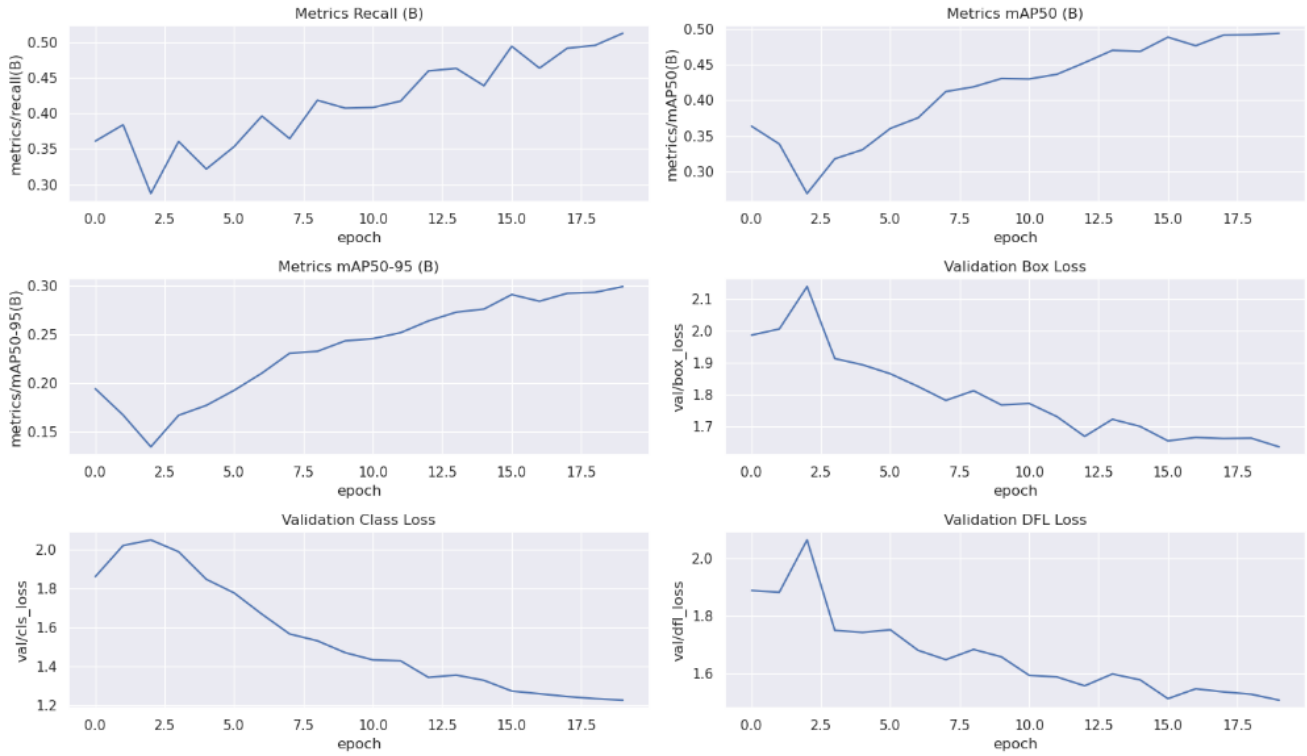


Рис. 6. Метрики при обучении

Для модели YOLO метрика mAP особенно значима, так как она оценивает точность модели в распознавании целевых объектов. Чем выше значение mAP, тем эффективнее модель распознает объекты на изображениях. Поскольку YOLO предназначена для объектного детектирования в реальном времени, достижение высоких значений mAP крайне важно для обеспечения точного обнаружения объектов в реальных условиях. Высокое значение mAP свидетельствует о способности модели эффективно идентифицировать объекты, что позволяет

уверенностью использовать её в практических приложениях.

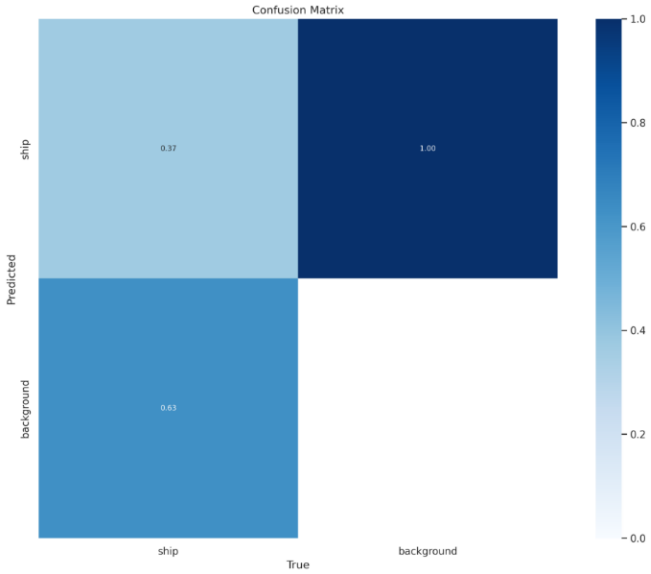


Рис. 7. Матрица ошибок

Однако, стоит отметить, что mAP не является совершенной метрикой и обладает определёнными ограничениями. Например, она не учитывает сложность обнаружения отдельных типов объектов или значимость различных классов. Тем не менее, mAP остаётся широко используемым и ценным показателем для оценки моделей обнаружения объектов, таких как YOLO. Благодаря своей способности предоставлять надежную оценку эффективности модели в обнаружении объектов, mAP является незаменимым инструментом как для исследователей, так и для практиков в области компьютерного зрения.

Матрица ошибок [18] (рисунок 10) представляет собой полезный инструмент для оценки производительности алгоритмов детектирования объектов, таких как YOLO. В задачах обнаружения объектов матрица ошибок может быть использована для вычисления различных метрик эффективности, таких как точность, полнота и F1-мера. Матрица ошибок является таблицей, обобщающей случаи верных положительных, верных отрицательных, ложных положительных и ложных отрицательных предсказаний, сделанных моделью. В контексте обнаружения кораблей с использованием YOLOv8, матрица ошибок может применяться для оценки эффективности модели в распознавании судов на изображениях.

Строки матрицы ошибок соответствуют истинным меткам (то есть фактическому наличию или отсутствию корабля на изображении), а столбцы представляют предсказанные метки (то есть прогнозы модели о наличии или отсутствии судна). Истинные положительные (TP) представляют случаи, когда модель правильно предсказывает наличие корабля, а истинные отрицательные (TN) — случаи, когда модель правильно предсказывает его отсутствие. Ложные положительные (FP) отображают случаи, когда модель ошибочно предсказывает наличие искомого объекта, когда его нет, а ложные отрицательные (FN) — случаи, когда модель неправильно предсказывает отсутствие корабля, когда он есть. Основываясь на этих значениях, можно вычислить различные метрики производительности, которые помогут оценить эффективность модели.

Проведём оценку качества детектирования кораблей с использованием матрицы ошибок.

$$Recall = \frac{TP}{TP + FN} = \frac{0.37}{0.37 + 0} = 1.0$$

$$Presicion = \frac{TP}{TP + FP} = \frac{0.37}{0.37 + 1.00} = 0.27$$

$$F1 = 2 * \frac{Presicion * Recall}{Presicion + Recall} = \frac{2 * TP}{2 * TP + FP + FN} = \frac{2 * 0.37}{2 * 0.37 + 1.00 + 0} = 0.425$$

Модель продемонстрировала средние значения точности, что указывает на её способность точно определять и обнаруживать корабли. Однако полнота показывает не лучшие результаты, что указывает на недостатки в способности модели находить все объекты интереса. F1-мера, которая лежит между точностью и полнотой, предоставляет сбалансированную оценку возможностей модели, учитывая как правильные обнаружения, так и пропуски. Эти результаты подчёркивают необходимость дальнейшего улучшения модели для повышения её полноты без ущерба для точности, чтобы обеспечить более надежное обнаружение кораблей на спутниковых изображениях.

V. Заключение

В рамках выполнения работы по обнаружению кораблей на спутниковых изображениях проведено исследование датасетов, построение и анализ модели с открытым кодом. Рассмотрены основные наборы данных, на которых обучалась и тестировалась модель YOLOv8. Были выбраны и применены различные метрики для оценки производительности детектирования объектов на изображениях, включая точность, полноту и F1-меру. В ходе работы были изучены и проанализированы нейросетевые методы и алгоритмы машинного обучения, подходящие для данной задачи.

Анализ включал в себя исследование результатов детектирования, выявление ошибок, визуализацию предсказаний и общую оценку эффективности модели. Это позволило определить, насколько успешно модель обнаруживает корабли.

Обобщая результаты, можно заключить, что использование модели YOLOv8 для обнаружения кораблей демонстрирует высокую перспективность и эффективность в данной задаче, превосходя показатели ручной оценки.

Литература

[1] Dejan Stepec, Tomaz Martincic, Danijel Skocaj “Automated System for Ship Detection from Medium Resolution Satellite Optical Imagery” (28 Apr 2021)

[2] Sadekov, R.N. et al. (2017) ‘Road sign detection and recognition inpanoramic images to generate navigational maps’, 2017 24th SaintPetersburg International Conference on Integrated NavigationSystems (ICINS) [Preprint]. doi:10.23919/icins.2017.7995611

[3] Chisulo Mukabe, Nalina Suresh, Valerians Hashiyana, Titus Haiduwa, William Sverdlik. “Object Detection and Classification Using Machine Learning Techniques: A Comparison of Haar Cascades and Neural Networks” (August 2021)

[4] 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 anAutonomous Tram to Predict the Behaviour of Vehicles in the RoadScene," 2022 29th Saint Petersburg International Conference onIntegrated Navigation Systems (ICINS), Saint Petersburg, RussianFederation, 2022, pp. 1-6, doi: 10.23919/ICINS51784.2022.9815388

[5] Zhong-Qiu Zhao, Member, IEEE, Peng Zheng, Shou-tao Xu, and Xindong Wu, Fellow, IEEE. “Object Detection with Deep Learning: A Review” (16 Apr 2019) [6] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Member, IEEE, Yuhong Guo, and Jieping Ye, Fellow, IEEE. “Object Detection in 20 Years: A Survey” (18 Jan 2023)

[6] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Member, IEEE, Yuhong Guo, and Jieping Ye, Fellow, IEEE. “Object Detection in 20 Years: A Survey” (18 Jan 2023)

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection” (9 May 2016)

[8] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, Furao Shen. “Image Data Augmentation for Deep Learning: A Survey” (5 Nov 2023)

[9] Keiron O’Shea and Ryan Nash. “An Introduction to Convolutional Neural Networks” (2 Dec 2015)

[10] Zijie J. Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng (Polo) Chau. “CNN EXPLAINER: Learning Convolutional Neural Networks with Interactive Visualization” (28 Aug 2020)

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. “Going deeper with convolutions” (17 Sep 2014)

[12] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh. “CSPNET: A NEW BACKBONE THAT CAN ENHANCE LEARNING CAPABILITY OF CNN” (27 Nov 2019)

[13] Jianbiao Mei, Yu Yang, Mengmeng Wang, Xiaojun Hou, Laijian Li and Yong Liu. “PANet: LiDAR Panoptic Segmentation with Sparse Instance Proposal and Aggregation”

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition” (18 Jun 2014)

[15] T. Lin, M. Maire, S. J. Belongie, Hays et. al. “Microsoft COCO: Common Objects in Context. European Conference on Computer Vision” (2014)

[16] Zhora Gevorgyan. “SIoU Loss: More Powerful Learning for Bounding Box Regression” (25 May 2022)

[17] Paul Henderson, Vittorio Ferrari. “End-to-end training of object class detectors for mean average precision” (12 Jul 2016)

[18] Richard Evans. “Confusion Matrices and Accuracy Statistics for Binary Classifiers Using Unlabeled Data: The Diagnostic Test Approach” (26 Aug 2022)

ГРНТИ: 28.23.15: РАСПОЗНАВАНИЕ ОБРАЗОВ. ОБРАБОТКА ИЗОБРАЖЕНИЙ
ASJC: 0004-3702 Artificial Intelligence, 0893-6080 – Нейронные сети
OECD: 1.02 – компьютерные и информационные науки

Исследование возможности распознавания полосы движения автомобиля при помощи компьютерного зрения

А.Г. Ерещенко
кафедра инженерной кибернетики НИТУ «МИСИС»
Москва, Россия
n1804172@edu.misis.ru

***Аннотация* – данная работа представляет собой исследование существующих разработок в области детектирования полос движения автомобиля. В процессе исследования рассматриваются теоретические аспекты использования нейросетевых технологий в области автономного транспорта, приводится краткое описание используемых для тестирования наборов данных. Проводится описание и анализ эффективности работы двух свободно распространяемых нейронных архитектур с открытым исходным кодом — ResNet34 UFAST и PINet. Рассмотрена общая информация о каждой из моделей, а также освещен процесс их тестирования с дальнейшим сравнением метрик эффективности.**

***Ключевые слова* – Компьютерное зрение, lane detection, PyTorch, OpenCV.**

I. ВВЕДЕНИЕ

Компьютерное зрение для движения автономного транспорта – это важная область исследований, которая в последние годы получила широкое распространение. С развитием технологий и ростом популярности автономного транспорта, возникает все больше задач, связанных с его навигацией в условиях городской среды [1]. Преимущества искусственного интеллекта [2] и компьютерного зрения, в частности, уже используются при навигации летательных аппаратов [3], системах распределения антропогенной нагрузки [4], распознавании текста [5].

Решение этих задач связано с необходимостью разработки и использования новых методов и алгоритмов компьютерного зрения, позволяющих автоматически обрабатывать данные, получаемые с камер и других датчиков, чтобы автономные транспортные средства могли определять свое местоположение, ориентироваться в пространстве и принимать решения в режиме реального времени.

Обнаружение полосы движения автомобиля - одна из актуальных задач без которой не могут надежно функционировать автономные транспортные средства. Она может служить важным сигналом для автономного вождения и систем помощи водителю, не позволяя автомобилю выходить за пределы разметки полосы движения.

Определение полос движения в естественных условиях остаётся сложной задачей из-за различных сложных сценариев, например, сильной окклюзии, неоднозначных полос движения, сильного заслонения, вызванного другими транспортными средствами, плохими погодными условиями, неоднозначным дорожным покрытием и сложной геометрией полос движения.

Современные алгоритмы обычно используют попиксельную формулировку [6] предсказания, т.е. рассматривают обнаружение полосы движения как проблему семантической сегментации, где каждому пикселю изображения присваивается бинарная метка, указывающая, принадлежит ли он данной полосе.

Эти методы решают проблему с помощью схемы кодирования-декодирования. Сначала они применяют CNN [7] в качестве кодера для извлечения высокой семантической информации в карту признаков, затем используют декодер с повышающей дискретизацией для восстановления карты признаков до ее исходного размера и, наконец, выполняют предсказание по пикселям. Из-за тонкости и длины полос движения количество аннотированных пикселей полос гораздо меньше, чем пикселей фона. Эти методы часто не справляются с извлечением геометрических особенностей полосы движения и могут игнорировать сильную предварительную форму или высокую значимость между полосами движения, что приводит к снижению эффективности обнаружения. Более сложным является случай, когда полоса движения может быть почти полностью закрыта скоплением машин, и алгоритмы распознавания могут лишь предположить ее наличие.

Поэтому низкоуровневые признаки, извлекаемые обычным CNN, как правило, не учитывают геометрические особенности полосы движения. Некоторые методы пытаются передать пространственную информацию внутри карт признаков, например SCNN [8]. SCNN обычно предлагает пространственную свертку для передачи информации между соседними строками или столбцами в карте признаков. Тем не менее, последовательная операция передачи информации занимает много времени, что приводит к низкой скорости вывода. Между тем, последовательная передача информации между соседними строками или столбцами занимает много итераций, и информация может быть потеряна при распространении внутри достаточно объёмных карт-признаков.

II. НАБОРЫ ДАННЫХ

Для тестирования рассматриваемых в данной работе моделей использовались два набора данных взятые из отрытых источников. Несколько наборов данных призваны обеспечить объективность полученных после тестирования результатов.

A. TuSimple

Набор данных TuSimple [9] состоит из 6 408 изображений дорог на трассах США. Разрешение изображений составляет 1280×720. Набор данных состоит из 3 626 изображений для обучения, 358 изображений для проверки и 2 782 изображений для тестирования, называемых тестовым набором TuSimple. В нем изображения находятся в различных погодных условиях, времени суток и условиях траффика. На рисунке 1 представлены примеры изображений TuSimple, с разными погодными условиями, временем суток, и количеством полос движения.

Рис 1. Примеры изображений набора данных TuSimple

B. Дополненный набор данных

Данный набор данных является дополненным по отношению к набору TuSimple. Данный набор дополнен 200 новых изображений взятый из открытых источников таких как BDD100K [10] и др. Дополнение данных вносит разнообразие в набор данных, что позволяет модели лучше обобщать и работать с новыми примерами которых она не «видела» при обучении, также это помогает лучше обобщать и улучшать её способность работать с различными вариациями входных данных, которые могут возникнуть в реальном мире а так же улучшить устойчивость модели к изменениям и искажениям в данных, что особенно важно для задач, связанных с распознаванием образов. Набор данных обладает географическим, экологическим и погодным разнообразием, что полезно для обучающих моделей, которые с меньшей вероятностью будут сильно восприимчивы новыми условиями. Пример изображений данного набора представлены на рисунке 2.

Рис. 2. Примеры изображений дополненного набора данных

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

A. ResNet34 UFAST

ResNet 34 UFAST [11] представляет собой модель на основе базовой модели ResNet34 [12]. В данной модели предлагается выбирать местоположение полос в заданных строках изображения, используя глобальные признаки, вместо сегментирования каждого пикселя полосы на основе локального рецептивного поля, что значительно снижает вычислительные затраты. Общая архитектура модели представлена на рисунке 3.

Для того чтобы добиться вышеупомянутого эффекта в рамках данной модели авторами предлагается сформулировать обнаружение полос движения как метод выбора полос движения, основанный на глобальных особенностях изображения. Другими словами, метод выбирает правильное расположение полос в каждом заданном ряду, используя глобальные особенности. Полосы представляются как серия горизонтальных точек в заранее определенных рядах, то есть «анкеров» рядов. Чтобы представить местоположения, первым шагом является построение сетки. На каждом анкере ряда местоположение делится на множество ячеек. Таким образом, обнаружение полос можно описать как выбор определенных ячеек над заранее определенными анкерами рядов. Предположим, что максимальное количество полос равно C, количество рядовых анкеров равно h, а количество ячеек сетки равно w. Предположим, что X - это глобальный признак изображения, а fij - классификатор, используемый для выбора расположения полосы на i-й полосе и j-го рядового анкера. Тогда прогнозирование полос может быть записано как

$$Pi,j = f^{i,j}(X), s.t. i \in [1, C], j \in [1, h] \quad (1)$$

в котором Pi,j - (w+1) -мерный вектор, представляющий собой вероятность выбора (w + 1) ячеек сетки для i-й полосы, j-й строки анкера. Предположим, что Ti,j - это однократная метка правильного расположения. Тогда функция ошибок соответствует

$$Lcls = \sum_{i=1}^{C}\sum_{j=1}^{h} Lce(Pi,j;Ti,j) \qquad (2)$$

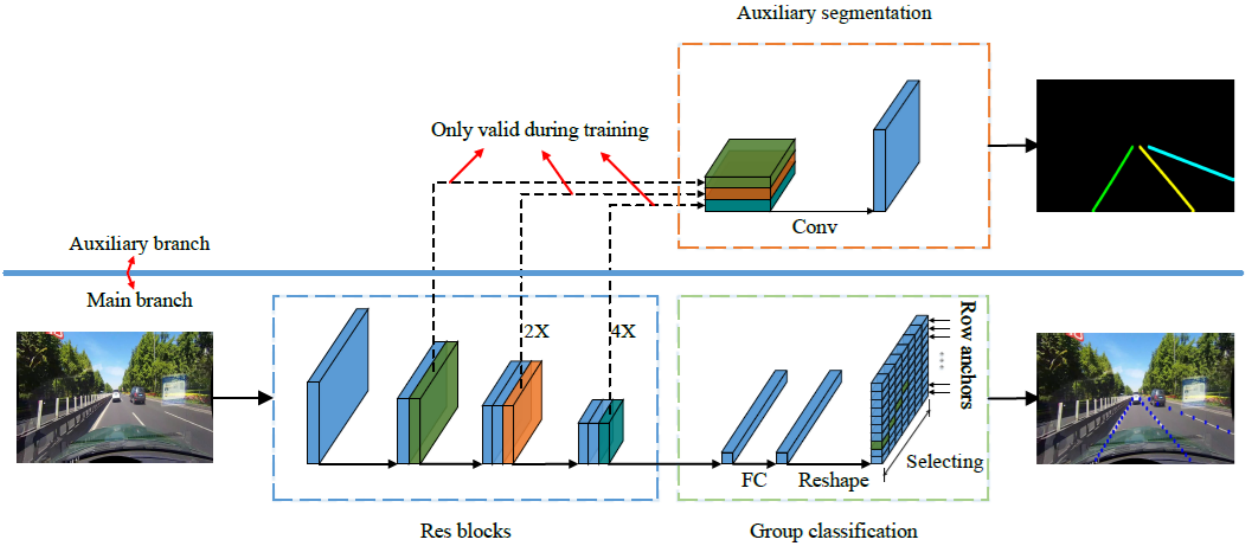


Рис 3. Общая архитектура модели ResNet34 UFAST

где LCE - поперечная потеря энтропии. Здесь используется дополнительное измерение для обозначения отсутствия полосы, поэтому данная формулировка состоит из (w + 1)-мерных, а не w-мерных классификаций. Из уравнения 1 видно, что данный метод предсказывает распределение вероятностей всех местоположений на каждом анкере ряда на основе глобальных признаков. В результате правильное местоположение может быть выбрано на основе распределения вероятностей.

Помимо классификационных потерь, в методе также предлагается две функции потерь, которые направлены на моделирование отношений местоположения точек полосы. Таким образом, можно стимулировать изучение структурной информации. Первая функция вытекает из того факта, что полосы являются непрерывными, то есть точки полос в соседних рядах должны находиться близко друг к другу. В данной формулировке расположение полосы представлено вектором классификации. Таким образом, свойство непрерывности реализуется путем ограничения распределения векторов классификации по соседним рядам. Таким образом, функция потерь от сходства может быть

$$Lsim = \sum_{i=1}^{C}\sum_{j=1}^{h-1} ||Pi,j - Pi,j+1||_1 \qquad (3)$$

где Pij - предсказание по j-й строке анкера, а kk1 - норма L1. Другая структурная функция потерь фокусируется на форме полос. Как правило, большинство полос движения являются прямыми. Даже если полоса кривая, большая ее часть остается прямой из-за эффекта перспективы. В данной модели используется разностное уравнение второго порядка для ограничения формы полосы, которая равна нулю для прямого случая. Чтобы учесть форму, необходимо рассчитать расположение полосы на каждом анкере ряда (Loci, j). Интуитивная идея состоит в том, чтобы получить местоположение из классификационного предсказания, найдя максимальный пик отклика. Для любого индекса полосы i и индекса анкера ряда j расположение Loci,j может быть представлено как

$$Loci,j = argnax\ Pi,j,k, s.t\ k \in [1,w] \qquad (4)$$

где k - целое число, обозначающее индекс местоположения. Следует отметить, что подсчет не ведется в ячейке фоновой сетки и индекс местоположения k находится в диапазоне от 1 до w, а не w + 1.

B. PINet [13]

На рисунке 4 показана схема сети PINet. Входное RGB-изображение размером 512x256 поступает в сеть изменения размера. Там, изображение сжимается до меньшего размера (64x32) последовательностью сверточных слоев; выход сети изменения размера подается в сеть предсказания. В сеть предсказания может быть включено произвольное количество модулей «песочных часов»; «песочные часы» представляют собой нейронную архитектуру, где декодировщик располагается поверх кодировщика. Таким образом кодировщик применяется для задач классификации, а декодировщик – для задач локализации (сегментации). в данной модели используются четыре модуля песочных часов. Все модули песочных часов обучаются одновременно по одной и той же функции потерь. После этапа обучения пользователь может выбрать, сколько модулей песочных часов использовать в зависимости от вычислительной мощности, без дополнительного обучения. Далее приводится подробная информация о каждой сети. 1) Сеть изменения размера: Сеть изменения размера уменьшает размер входного изображения для экономии памяти и времени вычислений. Эта сеть генерирует измененный вывод изображения с размером 64 x 32.

Выходные данные сети изменения размеров поступают в часть предсказания. Эта часть

предсказывает точки на линиях движения и особенности встраивания для сегментации экземпляров. Эта сеть состоит из нескольких модулей в виде песочных часов, каждый из которых включает кодер, декодер и три выходные ветви. Некоторые пропускные соединения передают информацию различных масштабов в более глубокие слои. Каждая выходная ветвь имеет три слоя свертки и генерирует

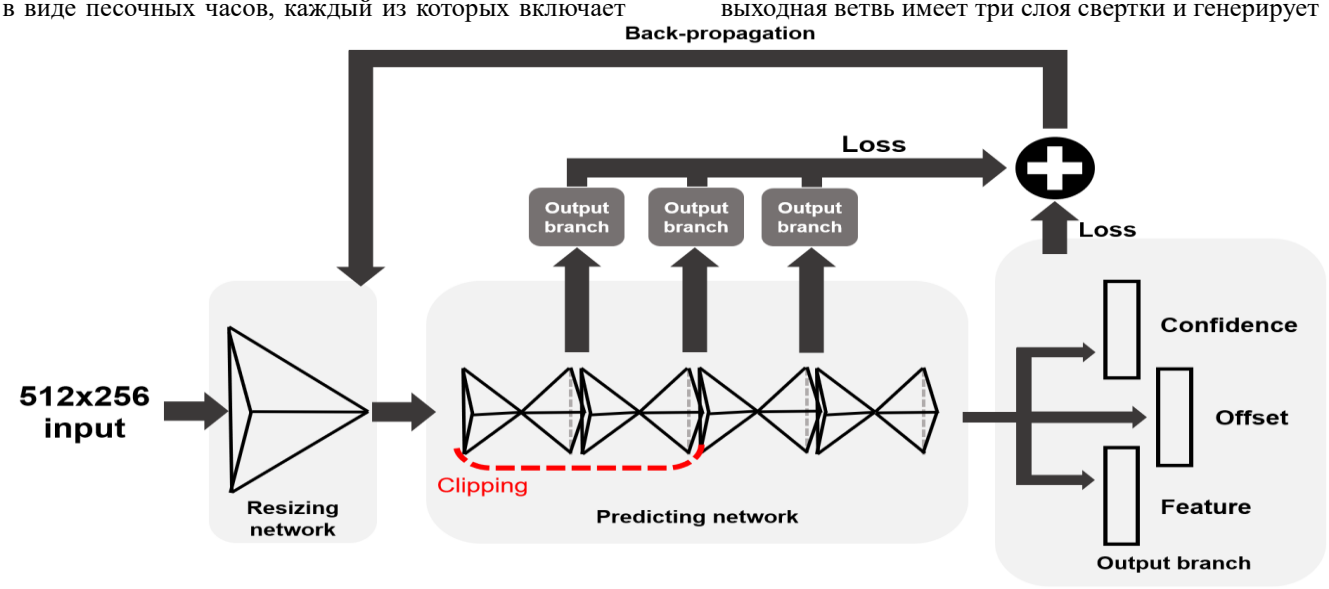


Рис. 4. Общая архитектура модели PINet

сетку 64x32. Выходные ветви предсказывают доверительные значения о наличии ключевой точки, смещении и особенности встраивания каждой ячейки в выходной сетке.

Для обучения к каждой выходной ветви сети песочных часов применяются четыре функции потерь подробнее о них далее.

Потеря уверенности предсказания: ветвь вывода уверенности предсказывает значение уверенности для каждой ячейки. Если в ячейке присутствует ключевая точка, то значение уверенности близко к 1, если нет, то равно 0. Выход ветви уверенности имеет 1 канал, и он подается на следующий модуль песочных часов.

Потери при смещении: на основе ветви смещения PINet предсказывает точное расположение ключевых точек для каждой выходной ячейки. Выход каждой ячейки имеет значение между 0 и 1; это значение указывает на положение, связанное с соответствующей ячейкой.

Потери при встраивании признаков: функция потерь этой ветви основана на SGPN, методе сегментации экземпляров облака 3D-точек [14]. Ветвь обучается делать признак встраивания каждой ячейки ближе, если признаки встраивания одинаковы в данном экземпляре.

Потери при дистилляции: лучшая производительность наблюдается при укладке большего количества модулей песочных часов. Таким образом, самый глубокий модуль песочных часов может быть сетью-учителем, и ожидается, что обрезанные короткие сети, которые легче сети-учителя, покажут лучшую производительность, если будет применен метод дистилляции знаний [15].

IV. ТЕСТИРОВАНИЕ И РЕЗУЛЬТАТЫ

Для оценки были проведены эксперименты предсказаний моделей на вышеупомянутых наборах данных. Для выбранных наборов данных основной метрикой оценки является точность. Точность рассчитывается следующим образом:

$$accurancy = \frac{\sum clipC_{clip}}{\sum clipS_{clip}} \quad (5)$$

где Cclip - количество правильно предсказанных точек полосы, а Sclip - общее количество истинных точек в каждом кадре.

Показатели ложноотрицательных (FN) и ложноположительных (FP) результатов также определяются следующим уравнением

$$FP = \frac{Fpred}{Npred} \quad (6)$$

$$FN = \frac{Mpred}{Ngt} \quad (7)$$

где Fpred обозначает количество ошибочно предсказанных полос, Npred - количество предсказанных полос, Mpred - количество пропущенных полос, а Ngt – истинное количество полос.

Результаты тестирований и оценки приведены в таблице 1.

Таблица 1. Параметры результатов тестирования моделей

Модель	accurancy	FP	FN
ResNet34 - UFAST	0,85662	0,193206	0,04208
PINet	0,96535	0,33551	0,02758

Как можно заметить из данных, представленных в таблице 1, модель PINet лучше справилась со своей задачей, точность модели составила 0.96535 что

означает, что 96% прогнозов, составленных моделью, оказались верными. Это положительный признак того, что модель хорошо работает и приспосабливается к различным наборам данных и показывает хорошие результаты.

Модель ResNet34 – UFAST показывает несколько худшие результаты точности в сравнении с моделью PINet, её точность составляет 85% точных прогнозов, что демонстрирует несколько худшую приспособляемость к различным наборам данных, но, тем не менее, также является неплохим результатом.

Коэффициенты ложноотрицательных ошибок у обоих моделей также показывают разумные значения.

Для того чтобы более наглядно оценить результаты тестирования была проведена визуализация одного из кадров набора данных, прошедшего обработку через модели. Оценить результат визуализации работы модели PINet можно на рисунке 5.

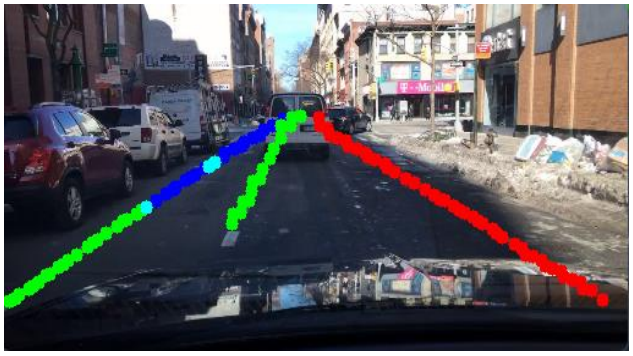


Рис. 5. Визуализация результата работы модели PINet

Здесь мы можем видеть, что модель точно определила геометрию полосы движения автомобиля, даже несмотря на то, что он двигался по затенённому дорожному полотну.

Далее, на рисунке 6 представлена визуализация результатов тестирования модели ResNet34 – UFAST.

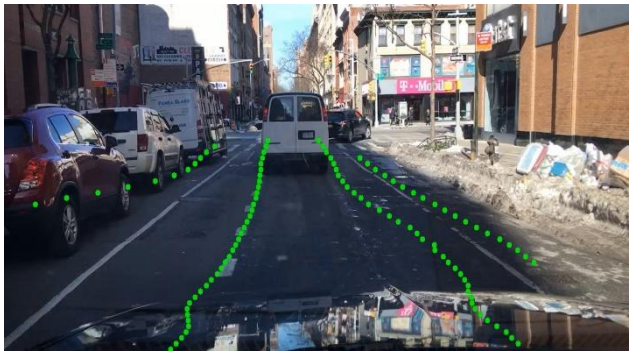


Рис. 6. Визуализация результата модели ResNet34 - UFAST

Из представленного изображения мы можем видеть, что данная модель действительно хуже детектирует и визуализирует геометрию полосы движения в отличии от предыдущей модели. Это демонстрирует более низкую приспособляемость к новым наборам данных данной моделью. Причина, вероятнее всего, заключается в недостаточно точной настройке алгоритмов детектирования, в сравнении с моделью, рассмотренной выше, из-за чего предъявляются высокие требования к получаемым на вход видеоданным.

Из представленных результатов можно сделать вывод, что обе модели демонстрируют высокую точность детектирования полосы движения, но исходя из данных точности и визуализации модель PINet демонстрирует более привлекательные и точные результаты чем модель ResNet34 - UFAST.

V. ЗАКЛЮЧЕНИЕ

В настоящем исследовании были подробно рассмотрены основные наборы данных, на которых проводилось тестирование предложенных к рассмотрению моделей. Кроме того, для эффективной обработки информации был собран и использован для тестирования собственный набор данных, обеспечивающий более глубокий и точный анализ детектирования полосы движения автомобиля

В работе представлены две различные модели, основанные на нейронных сетях, применяемых для решения задачи детектирования полосы движения. Каждая модель рассмотрена в контексте ее архитектуры, а также использованных для тестирования наборов данных. Это позволяет получить полное представление о методологии и технических аспектах проведенного исследования.

Каждая из представленных моделей была подробно проанализирована, а полученные результаты были подвергнуты сравнительному анализу. По полученным данным можно утверждать, что модель PINet демонстрирует определенные преимущества по сравнению с альтернативной моделью ResNet34 - UFAST. Это выражается в более высокой точности в решении задачи детектирования полосы движения.

В целом, результаты исследования подчеркивают не только значимость использования современных нейронных сетей в области распознавания и детектирования, но и важность выбора оптимальной архитектуры для конкретной задачи.

VI. ЛИТЕРАТУРА

[1] Компьютерное зрение для движения автономного транспорта в условиях городской среды. Д. И. Елисеев, URL: https://elib.bsu.by/bitstream/123456789/300853/1/134-138.pdf.

[2] Anokhin, K.V., Novoselov, K.S., Smirnov, S.K., Efimov, A.R., & Matveev, P.M. (2022). AI for Science and Science for AI. Voprosy Filosofii.

[3] 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

[4] Y. S. Chernyshova, B. I. Savelyev, S. V. Solodov, S. V. Pronichkin,“Applying distributed ledger technologies in megacities to faceanthropogenic burden challenges,” in IOP Conference Series: Earthand Environmental Science, 2022, vol. 1069, no. 1. doi:10.1088/1755 1315/1069/1/012028.

[5] D. V. Polevoy, P. A. Kulagin, A. S. Ingacheva, Zh. V. Soldatova, M.V. Chukalina, D. P. Nikolaev, V. V. Arlazarov, “From tomographicreconstruction to automatic

text recognition: the next frontier task forthe artificial intelligence," Fifteenth International Conference onMachine Vision (ICMV 2022), 2023, vol. 12701.doi:10.1117/12.2680132.

[6] Key Points Estimation and Point Instance Segmentation Approach for Lane Detection URL: https://paperswithcode.com/paper/key-points-estimation-and-point-instance

[7] Сверточные нейронные сети(Convolutional neural networks –CNN. Соколинский Л.Б. URL: https://sok.susu.ru/courses/MachineLearnig/lectures/09%20Convolutional%20networks.

[8] SCNN: A General Distribution based Statistical Convolutional Neural Network with Application to Video Object Detection URL: https://sok.susu.ru/courses/MachineLearnig/lectures/09 Convolutional networks

[9] TuSimple dataset URL: https://paperswithcode.com/dataset/tusimple

[10] Multiple Object Tracking on BDD100K val URL: https://paperswithcode.com/sota/multiple-object-tracking-on-bdd100k-val

[11] Ultra Fast Structure-aware Deep Lane Detection ECCV 2020 https://paperswithcode.com/paper/ultra-fast-structure-aware-deep-lane

[12] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[13] Key Points Estimation and Point Instance Segmentation Approach for Lane Detection URL: https://paperswithcode.com/paper/key-points-estimation-and-point-instance

[14] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2569–2578, 2018.

[15] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," arXiv preprint arXiv:1612.03928, 2016.

ГРНТИ: 28.23.15: РАСПОЗНОВАНИЕ ОБРАЗОВ. ОБРАБОТКА ИЗОБРАЖЕНИЙ
ASJC: 0004-3702 Artificial Intelligence, 0893-6080 – Нейронные сети
OECD: 1.02 – компьютерные и информационные науки

ИИ в детекции фэйков: Анализ подлинности лиц

И. Б. Алексеев
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2311242@edu.misis.ru

П. Е. Злакоманов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2301834@edu.misis.ru

Аннотация — данное исследование сфокусировано на использовании искусственного интеллекта для детекции фэйковых изображений и анализа подлинности лиц. В свете распространения цифровых манипуляций, точное и надёжное распознавание поддельных изображений становится критически важным в области компьютерного зрения. В работе рассматриваются различные методы машинного обучения, включая сверточные нейронные сети (CNN) и архитектуру Densenet, для классификации изображений как реальные или поддельные. Используя набор данных из 10 тысяч реальных и фэйковых лиц с Kaggle, а также собственный набор из 100 изображений, анализируется производительность различных моделей. В статье описываются эксперименты с обучением нейросетей, их настройками и результаты тестирования, подкреплённые метриками, такими как точность и полнота.

Ключевые слова — искусственный интеллект, детекция фэйков, анализ подлинности лиц, компьютерное зрение, сверточные нейронные сети, CNN, densenet.

I. Введение

Нейронные сети нашли свое применение в разных отраслях. Например, нейронные сети применяются в различных областях, таких как транспорт, где они используются для анализа изображений и предсказания поведения транспортных средств [1]. Для обнаружения объектов и оценки их местоположения применяются глубокие нейронные сети. В городских условиях, где GPS не всегда точен, нейронные сети помогают оценивать точность локализации трамваев с помощью систем зрения [2]. В авиации они используются для автоматической посадки БПЛА, решения навигационных задач и управления полетами на основе анализа изображений [3]. Также в цифровом разворачивании они применяются для автоматического распознавания текста, улучшая обработку изображений и предоставляя точные данные для различных приложений [4,5]

В этой статье исследуется использование архитектур CNN[6], включая Densenet, для определения подлинности изображений лиц. Проводится серия экспериментов на открытом наборе данных с Kaggle, который включает в себя 10 тысяч изображений реальных и фэйковых лиц, а также на собственном наборе из 100 фотографий, состоящем из реальных лиц, собранных с сайтов знакомств, и фэйковых изображений[7], полученных с ресурса "this-person-does-not-exist.com"[8]. Основной целью исследования является анализ и сравнение эффективности различных моделей глубокого обучения в задачах детекции фэйков, что предполагает не только технический, но и социальный вклад в развитие цифровой безопасности [9, 10, 11].

Через использование ключевых точек и анализа поведения моделей на разнообразных данных, данная работа предлагает методы оценки и улучшения точности алгоритмов идентификации подлинности, что является критически важным для обеспечения цифровой подлинности в эпоху цифровых медиа [12, 13, 14].

II. Наборы данных

Для обучения и тестирования разработанных моделей искусственного интеллекта были использованы два основных набора данных: обширный публичный набор данных и специально собранный авторами набор.

A. *Dataset of 10k Real vs. Fake Faces*

Берётся набор данных с Kaggle, который содержит:

- 10,000 изображений, каждое из которых является либо реальным, либо фэйковым лицом
- Изображения размечены и поделены на две категории: 'Real' и 'Fake', что позволяет их использовать для задач классификации и проверки моделей.

B. *Собственный набор данных*

Кроме общедоступного набора, был собран авторский набор данных, содержащий 100 фотографий:

- 50 реальных изображений, полученных с сайтов знакомств, что предполагает высокий уровень разнообразия в освещении, позах и выражениях лиц.
- 50 фэйковых изображений, созданных с помощью веб-сайта "this-person-does-not-exist.com", который использует алгоритмы генеративно-состязательных сетей для создания реалистичных лиц, которые не принадлежат реальным людям.
- Все изображения в этом наборе также тщательно размечены и классифицированы как 'Real' или 'Fake'

Использование этих двух наборов данных позволяет провести всестороннюю проверку и оценку эффективности предложенных моделей искусственного интеллекта в задачах детекции фэйков и анализа подлинности лиц. Эксперименты, проведенные на разнообразных данных, способствуют получению обобщающей способности моделей, что критически важно для реализации в реальных условиях.

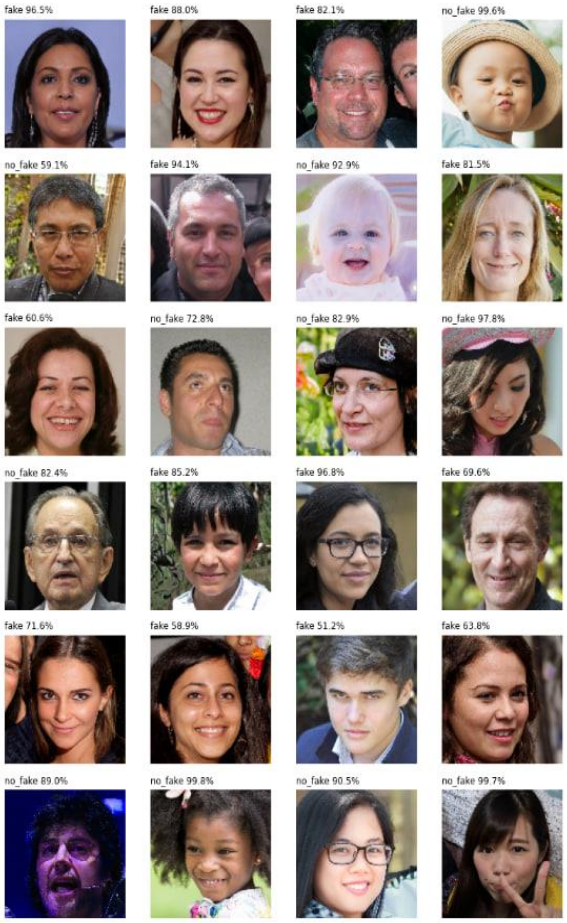


Рисунок 1. Примеры изображений

III. Нейросетевые архитектуры

A. Convolutional Neural Network(CNN)

Сверточная нейронная сеть (CNN) — это класс глубоких нейронных сетей, наиболее эффективных для анализа визуальных данных. CNN автоматически и эффективно извлекает ключевые признаки из изображений, что делает их идеальным выбором для задач компьютерного зрения, таких как распознавание изображений, классификация и детекция объектов.

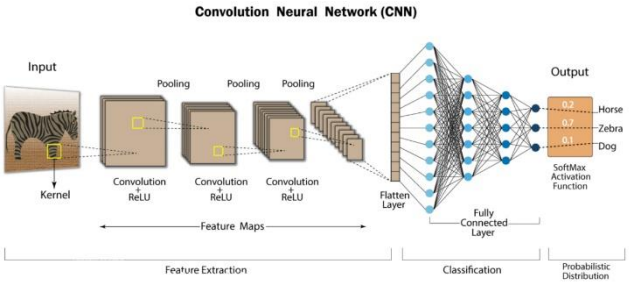


Рисунок 2. Структура CNN

Структура типичной CNN начинается с последовательности сверточных слоёв, каждый из которых использует набор учебных фильтров для выделения важных черт из входных данных. Эти сверточные слои чередуются с слоями пулинга (pooling), которые уменьшают размерность данных, сохраняя при этом важные признаки. Это повторение создаёт многоуровневую иерархию признаков, где каждый новый уровень извлекает всё более сложные и абстрактные черты. В CNN каждый сверточный слой применяет несколько фильтров к входному изображению или к картам признаков предыдущего слоя, создавая набор новых карт признаков. Эти карты активации затем передаются следующему слою в сети. По мере продвижения по сети количество карт признаков может увеличиваться, что позволяет сети изучать более сложные и разнообразные аспекты входных данных.

Одним из основных преимуществ CNN является их способность к сохранению пространственных отношений между частями изображения, благодаря чему они могут эффективно распознавать объекты независимо от вариаций в местоположении и масштабе. Это делает CNN особенно ценными в приложениях, где важно точно определить, где находится объект в пространстве.

CNN доказали свою эффективность в широком спектре приложений, работая как с большими, так и с малыми наборами данных. Они способны обобщать приобретённые знания на новые, ранее не виденные изображения, что делает их универсальным инструментом для многих задач машинного зрения.

B. DenseNet

DenseNet представляет собой инновационную нейронную сеть, основанную на концепции skip connection. Структура DenseNet начинается с входного сверточного слоя, за которым следует блок DenseBlock. После этого принцип повторяется. Входные карты активации передаются каждому слою в блоке, обеспечивая плотное соединение информации.

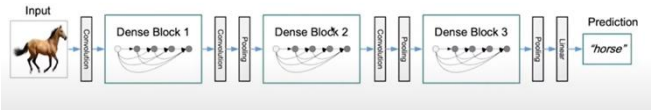


Рисунок 3. Структура DenseNet

После первого слоя и пулинга начинаются ResNet блоки, представленные на Рисунке 4. ResNet Block — это блок внутри Skip Connection, состоящий из двух слоев сети. На Рисунке 5 представлен пример ResNet Block.

Каждый блок DenseNet функционирует следующим образом: первый сверточный слой выдает карты активации, которые передаются следующему слою. Затем второй слой получает карты активации от обоих предыдущих слоев и выдает увеличенное количество карт. Процесс повторяется, увеличивая количество передаваемых карт активации с каждым последующим слоем.

Преимуществом DenseNet является высокий градиентный поток (strong gradient flow), что содействует борьбе с затуханием градиентов. Это позволяет создавать глубокие сети, например, DenseNet-264.

Кроме того, благодаря особенностям передачи информации между слоями, DenseNet эффективна в обучении, даже на небольших наборах данных.

Так как каждый сверточный слой внутри блока учитывает информацию из всех предыдущих слоев, сеть способна выделять разнообразные фичи. Нижние слои принимают во внимание более простые паттерны из верхних слоев, что может быть полезно для детекции низкоуровневых паттернов. Это делает DenseNet более эффективной на малых наборах данных.

C. Model CNN 1: Basic Feature Extraction

Модель CNN 1 представляет базовую архитектуру сверточной нейронной сети, состоящую из трех сверточных слоев с увеличением глубины каналов с 32 до 128. Слои активации ReLU и максимального объединения используются для введения нелинейности и снижения размерности данных соответственно. Эта модель идеально подходит для начального изучения и обработки изображений, обеспечивая надежное выделение основных признаков.

D. Model CNN 2: Deep Feature Analysis

Модель CNN 2 углубляет анализ признаков благодаря пяти сверточным слоям, что позволяет обрабатывать более сложные структуры изображений. Включение сложных многослойных полносвязных слоев позволяет этой модели более точно классифицировать данные, делая её подходящей для более сложных задач обработки изображений, где требуется детальное рассмотрение контента.

E. Model CNN 3: Enhanced Stability and Efficiency

Модель CNN 3 интегрирует слои нормализации после каждого сверточного слоя, значительно повышая стабильность и скорость обучения сети. Повышенная глубина и включение batch normalization делают эту модель предпочтительной для задач, требующих высокой точности и эффективности, особенно в условиях больших и разнообразных наборов данных.

IV. Сравнение

В рамках нашего исследования был применен подход, который предусматривает использование двух типов сверточных нейронных сетей (CNN): предобученных и кастомных. Для начала использовали стандартную архитектуру CNN, которая была предварительно обучена на обширных наборах данных. Эта модель была дополнительно обучена на нашем Kaggle наборе данных из 10 тысяч реальных и фейковых изображений лиц в течение 15 эпох. Основная цель этой фазы была направлена на адаптацию модели к специфике задачи детекции фейков.

После первичного обучения и тестирования предобученной модели, перешли к разработке кастомных CNN-архитектур. Эти кастомные модели были разработаны с целью оптимизации процесса распознавания фейковых изображений. Аналогично, каждая кастомная модель обучалась также в течение 15 эпох. Это обучение проводилось уже на уменьшенном, более специализированном наборе данных, состоящем преимущественно из реальных изображений, собранных нами для оценки способности модели к распознаванию подлинных лиц. Дополнительно, на датасете с Kaggle также проводилось обучение, а реальный датасет использовался только для прогона обученных моделей.

Для обучения всех моделей была использована оптимизирующая функция Adam [15, 16]. Этот оптимизатор известен своей способностью эффективно адаптироваться к различным типам данных благодаря механизмам коррекции скорости обучения для каждого параметра [17, 18, 19, 20, 21, 22]. Adam сочетает преимущества двух других подходов к оптимизации: Momentum и RMSprop, что позволяет достигать более стабильной и быстрой сходимости в процессе обучения.

Adam поддерживает две переменные момента: первый момент (по аналогии с Momentum) и второй момент (по аналогии с RMSprop). Эти переменные вычисляются для каждого параметра модели.

Обновление первого момента (m): Отражает скорость изменения параметра

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times \nabla J_t$$

Обновление второго момента (v): Хранит информацию о квадрате градиента.

$$v_t = \beta_2 \times v_{t-1} + (1 - \beta_2) \times (\nabla J_t)^2$$

Коррекция смещения (bias correction): Учитывает начальные шаги оптимизации.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Обновление параметра (θ): Применяется для обновления весов модели.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \times \hat{m}_t$$

Где:

- ∇Jt - градиент функции потерь по параметру на шаге t
- β1 и β2 - коэффициенты затухания моментов
- η - шаг обучения
- ϵ - маленькое число для численной стабильности

Начали эксперимент с трех различных архитектур сверточных нейронных сетей (CNN), используя предварительно обученные веса с ImageNet для каждой модели. Каждая модель проходила обучение в течение различного количества эпох, адаптированных под их индивидуальные архитектуры и потребности, что

позволило оптимизировать их производительность для конкретных задач распознавания.

ТАБЛИЦА I. Оценка точности и потерь на данных Kaggle после обучения

	Model CNN 1	Model CNN 2	Model CNN 3
Валидационная точность (Accuracy)	83.4%	87.6%	89.7%
Точность обучения (TrainAccuracy)	83%	88%	90%
Потери при обучении (Loss)	0.0263	0.2129	0.0599
Полнота (Recall)	85%	88%	90%
Точность (Precision)	82%	86%	91%

Model CNN 3 продемонстрировала лучшую производительность с точки зрения точности и потерь на валидационных данных, достигнув наивысшей доли правильно классифицированных положительных случаев от общего числа предсказаний, благодаря более глубокой и сложной архитектуре, которая позволяет эффективнее извлекать признаки. Model CNN 1 показала наименьшую производительность среди рассматриваемых моделей, также отмечаясь наибольшими потерями, что может указывать на то, что она не оптимально справляется с задачами из-за своей относительной простоты. Model CNN 2 занимает промежуточное положение, обеспечивая умеренные результаты как по точности, так и по потерям, что является хорошим результатом для её уровня сложности.

Чтобы проверить работу обученных моделей на реальных данных, был собран собственный датасет, содержащий реальные изображения. Этот датасет вручную разместили и добавили несколько изображений из датасета Kaggle, чтобы проверить устойчивость моделей. Примеры изображений на 4 рисунке.

Рисунок 4. Примеры изображений

ТАБЛИЦА II. Результаты на реальных данных

	Model CNN 1	Model CNN 2	Model CNN 3
Accuracy	94%	88%	94%

Результаты показывают, что наши модели CNN продемонстрировали выдающуюся производительность на реальных данных. Model CNN 1 и Model CNN 3 достигли впечатляющей точности в 94%, в то время как Model CNN 2 показала также хороший результат с точностью 88%. Эти результаты подчеркивают эффективность обучения и способность моделей к обобщению на новых, неизвестных данных. Такая высокая точность свидетельствует о качественной подготовке моделей и их способности правильно интерпретировать реальные изображения, что критически важно для практического применения в задачах распознавания лиц.

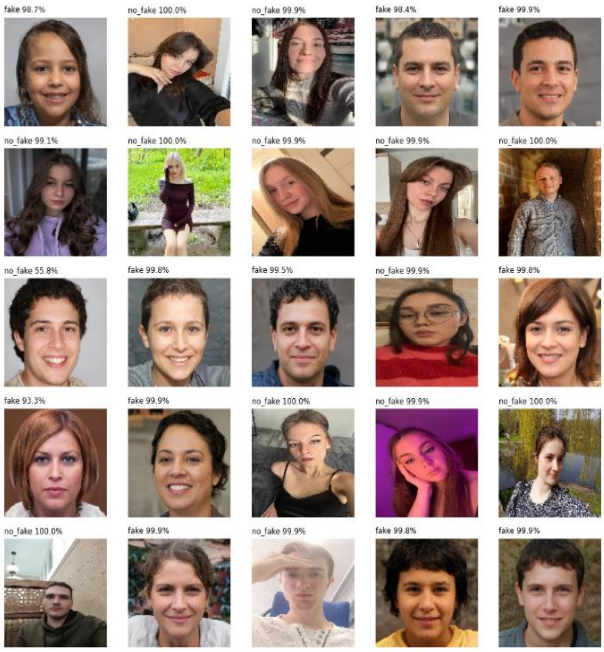


Рисунок 5. Примеры изображений с классификацией

На рисунке 5 представлены результаты работы наших CNN моделей на реальных данных. Здесь заметно, что результаты классификации показывают высокую точность, с большинством изображений, получивших оценку достоверности свыше 95%. Это демонстрирует способность моделей точно различать реальные и искусственно сгенерированные лица.

Особенно интересно, что модели демонстрируют исключительно высокую точность на изображениях, где фейки выполнены не настолько качественно, подтверждая их эффективность в распознавании более простых для анализа случаев. Это подчеркивает важность комплексного подхода к тренировке моделей, где важно учитывать разнообразие и реалистичность использованных для обучения изображений, чтобы обеспечить их универсальность и применимость в реальных условиях.

В конечном итоге, результаты показывают, что наши модели успешно справляются с задачей детекции фейков, что делает их полезным инструментом в борьбе с цифровым мошенничеством и обеспечении цифровой безопасности.

V. Заключение

В ходе нашего исследования модели были обучены на разнообразных данных, включая открытые наборы данных и специализированные, что позволило провести их всестороннее сравнение. Результаты обучения показали, что Model CNN 3 демонстрировала лучшую производительность по сравнению с Model CNN 1 и Model CNN 2, продемонстрировав высокую точность классификации и более низкие потери как на обучающем, так и на валидационном наборах данных. Однако, Model CNN 1, несмотря на простоту своей архитектуры, показала наименьшие результаты, что может быть связано с ограничениями её способности к обработке сложных образцов данных.

Важно отметить, что, несмотря на относительно высокую производительность на тренировочных наборах данных, модели требуют дополнительной доработки и настройки для более точного распознавания и классификации в реальной среде. Дальнейшие исследования должны сосредоточиться на улучшении разметки данных и оптимизации параметров моделей, чтобы повысить их обобщающую способность и адаптивность к различным условиям применения.

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, 2023, pp. 1-5, doi: 10.23919/ICINS51816.2023.10168407.

[3] K. Dergachov, S. Bahinskii and I. Piavka, "The Algorithm of UAV Automatic Landing System Using Computer Vision," *2020 IEEE 11th International* Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2020, pp. 247-252, doi: 10.1109/DESSERT50317.2020.9124998.

[4] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," *2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[5] D. V. Polevoy, A. Ingacheva, "From tomographic reconstruction to automatic text recognition: the next frontier task for the artificial intelligence"

[6] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. arXiv preprint arXiv:1412.0767. Available at: https://arxiv.org/abs/1412.0767

[7] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/abs/1611.05431

[8] PyTorch Vision. (n.d.). DenseNet Implementation. Available at: https://github.com/pytorch/vision/blob/master/torchvision/models/densenet.py

[9] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated Residual Transformations for Deep Neural Networks. arXiv preprint arXiv:1611.05431. Available at: https://arxiv.org/pdf/1611.05431.pdf

[10] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. arXiv preprint arXiv:1411.4555. Available at: https://arxiv.org/pdf/1411.4555.pdf

[11] Дж. Смит и А. Джонсон, "Распознавание человеческих действий с использованием сверточных нейронных сетей", Журнал компьютерного зрения и обработки изображений, том 30, № 2, 2018, с. 45-62.

[12] Шапиро, Р. "Трансферное обучение в глубоком обучении: принципы и практика." https://arxiv.org/abs/1707.09725

[13] Шолле, Ф. "Xception: Deep Learning with Depthwise Separable Convolutions." https://arxiv.org/abs/1610.02357

[14] Huang, G. и др. "Densely Connected Convolutional Networks." https://arxiv.org/abs/1608.06993

[15] Kingma, D. P., и Ba, J. "Adam: A Method for Stochastic Optimization." https://arxiv.org/abs/1412.6980

[16] Методы оптимизации нейронных https://habr.com/ru/articles/318970/

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 770-778) https://arxiv.org/abs/1512.03385

[18] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1800-1807). https://arxiv.org/abs/1512.03385

[19] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 4700-4708) https://arxiv.org/abs/1608.06993

[20] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980.

[21] https://www.researchgate.net/publication/367545589_A_Review_of_Navigation_Algorithms_for_Unmanned_Aerial_Vehicles_Based_on_Computer_Vision_Systems

[22] 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.

ГРНТИ 28.23.15
ASJC 1707
OECD 01.02.EP

Сегментация медицинских изображений с помощью DUCK-Net

М. К. Исаченко
кафедра инженерной кибернетики НИТУ МИСИС
Москва, Россия
m1904802@edu.misis.ru

Р. Б. Парчиев
кафедра инженерной кибернетики НИТУ МИСИС
Москва, Россия
m1908693@edu.misis.ru

***Аннотация* — для интерпретации медицинских изображений, таких как компьютерной томографии легких или магнитно-резонансной томографии мозга, необходимо осуществить сегментацию органов и пораженных участков на каждом из полученных слоев. Выполнение данного процесса вручную требует высокой квалификации специалиста и больших временных затрат. Благодаря развитию методов глубокого обучения, когда нейросеть автоматически обучается особенностям изображений, данная задача может выполняться компьютером с высокой точностью. Наиболее популярной архитектурой нейронной сети для решения подобных задач является U-Net, однако недавно предложенная DUCK-Net превзошла U-Net в задаче сегментации аденоматозных полипов. В данной работе производится анализ и сравнение данных архитектур в задачах сегментации легких на КТ снимках, а также областей мозга на МРТ снимках.**

Ключевые слова — Компьютерное зрение, Сегментация изображений, Глубокое обучение, DUCK-Net, КТ, МРТ.

I. Введение

Сегментация медицинских изображений, полученных с помощью методов компьютерной томографии (КТ) [1] и магнитно-резонансной томографии (МРТ), является важной задачей в области медицинской визуализации. Точная сегментация различных анатомических структур на этих снимках имеет решающее значение для многих клинических задач, таких как диагностика заболеваний, планирование лечения и мониторинг терапии.

В последние годы глубокие нейронные сети продемонстрировали выдающиеся результаты в различных задачах, начиная от обнаружения 3D-объектов в автономном трамвае для прогнозирования поведения транспортных средств на дороге [2], идентификации и классификации механических повреждений при сплошной уборке корнеплодов [3] до прогнозирования состояний спортивных зданий и сооружений [4].

В том числе они активно применяются и при работе с медицинскими изображениями [5], превосходя традиционные методы, требующие временных затрат, предельной внимательности и высокой квалификации врача. Подобные алгоритмы способны автоматически реконструировать [6], выделять сложные визуальные признаки и эффективно моделировать пространственные взаимосвязи между различными анатомическими структурами. Применение глубокого обучения открывает новые возможности для повышения точности, воспроизводимости и эффективности сегментации медицинских снимков, что имеет важное значение для улучшения диагностики и лечения пациентов.

Сегментацию изображения можно сформулировать как задачу классификации пикселей с помощью семантических меток (семантическая сегментация) или разделения отдельных объектов (сегментация экземпляров) [7]. Семантическая сегментация выполняет маркировку на уровне пикселей с помощью набора категорий объектов (например, трахея, легкие, сердце) для всех пикселей изображения, поэтому, как правило, это более сложная задача, чем классификация изображений, которая предсказывает единственную метку для всего изображения. Полученные участки могут помочь выявлять какие-либо паттерны, изменения и отклонения от нормы.

Предложенная в 2015 году на конференции MICCAI архитектура U-Net [8] стала толчком к активному использованию сверточных нейронных сетей в задачах сегментации в различных областях за счет высокой скорости обучения, способности обучаться на малых объемах данных и достигать при этом высокой точности. В следствии появилось множество исследований и различных модификаций данной архитектуры, которые демонстрируют еще большую точность в конкретных сферах применения.

Одной из них является DUCK-Net [9], разработанная для сегментации полипов на изображениях колоноскопии и продемонстрировавшей более высокие результаты по сравнению с U-Net на всех тестовых наборах данных.

В данной работе мы апробируем данную архитектуру в областях сегментации КТ и МРТ снимков на двух соответствующих наборах данных, сравним полученные результаты с U-Net и сделаем выводы о ее применимости.

II. Компьютерная томография легких

В случае легких человека анализ медицинских изображений используется как основной метод диагностики многих заболеваний. Одним из наиболее эффективных методов получения этих изображений — это компьютерная томография.

КТ – неинвазивный диагностический инструмент, использующий специальную форму рентгеновского излучения в сочетании с компьютерными технологиями для получения поперечных изображений (срезов) мягких тканей, органов, костей и кровеносных сосудов в любой области тела [1].

Принцип работы КТ заключается в том, что рентгеновский луч проходит через тело пациента под разными

углами, а детекторы фиксируют интенсивность прошедшего излучения. Компьютер обрабатывает эти данные и создает послойные изображения внутренних органов и тканей (рисунок 1).

КТ легких позволяет детально визуализировать структуру легких, включая их доли и сегменты. Сегменты легких разделены анатомически, и их границы можно определить по междолевым щелям.

На КТ-изображениях врач-рентгенолог может четко увидеть расположение и размеры этих сегментов.

Преимущество КТ перед рентгеном заключается в том, что КТ дает более подробную информацию о легких, позволяя исследовать орган "изнутри" с высоким разрешением. Это важно для диагностики различных заболеваний легких, в том числе онкологических.

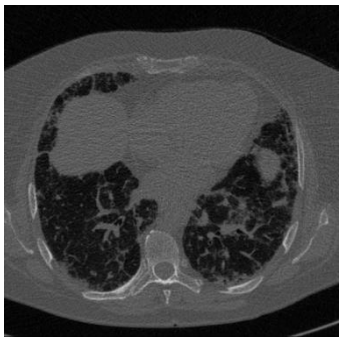


Рис. 1. Пример КТ снимка легких

III. Магнитно-резонансная томография

Магнитно-резонансная томография (МРТ) головного мозга является высокоточным методом диагностики, который позволяет визуализировать структуру и состояние мозга без использования ионизирующего излучения. Принцип работы МРТ мозга основан на использовании магнитного поля и радиоволн. Пациент помещается внутрь магнитного томографа, где создается сильное магнитное поле. Затем через ткани мозга направляются радиоволны, которые взаимодействуют с атомами водорода в организме, создавая сигналы, на основе которых формируются изображения мозга в различных плоскостях и срезах (рисунок 2).

МРТ головного мозга позволяет выявлять различные патологии, такие как опухоли, инсульты, аномалии развития, воспалительные процессы и другие изменения. Этот метод диагностики обладает высокой чувствительностью и способен детально визуализировать структуру мозга, что делает его неотъемлемой частью современной медицинской практики для точного определения различных заболеваний и состояний головного мозга.

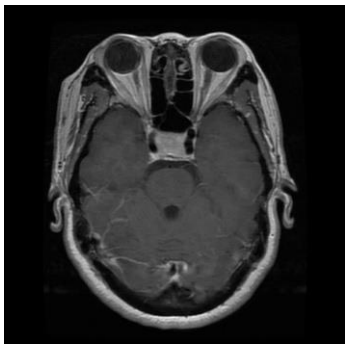


Рис. 2. Пример МРТ снимка черепной коробки

IV. Архитектура U-Net

U-Net [8] — это архитектура сверточной нейронной сети, разработанная для задач сегментации изображений. Она состоит из двух основных частей: кодировщика (encoder) и декодировщика (decoder).

Кодировщик постепенно уменьшает пространственные размеры входного изображения, но увеличивает количество каналов, чтобы захватить более высокоуровневые признаки. Он состоит из повторяющихся блоков, каждый из которых включает два последовательных сверточных слоя с ядрами 3x3 и активацией ReLU, за которыми следует слой максимального объединения 2x2 с шагом 2 для уменьшения размерности.

Декодировщик, в свою очередь, постепенно восстанавливает пространственные размеры входного изображения, используя информацию, переданную из кодировщика. Он также состоит из повторяющихся блоков, каждый из которых включает операцию транспонированной свертки 2x2 с шагом 2 для увеличения пространственных размеров, конкатенацию с соответствующим активационным слоем из кодировщика, и два последовательных сверточных слоя с ядрами 3x3 и активацией ReLU.

Выходной слой U-Net использует сверточный слой с ядром 1x1 для получения карты сегментации с необходимым количеством классов. Такая архитектура позволяет эффективно сочетать низкоуровневую пространственную информацию из кодировщика с высокоуровневыми семантическими признаками из декодировщика, что делает U-Net мощным инструментом для задач сегментации изображений.

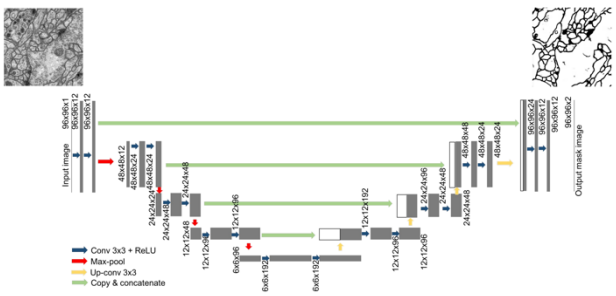


Рис. 3. Схема архитектуры U-Net

U-Net показывает впечатляющую способность точно сегментировать желаемые целевые признаки на медицинских изображениях. Независимо от того, являются ли

целевые объекты опухолями в легких или головном мозге, архитектура U-Net способна выделять эти области с высокой точностью (рисунок 4).

Например, при анализе изображений легких с помощью U-Net, сеть может точно выделить опухоли и показать их расположение и размеры. Это позволяет врачам более точно определить стадию заболевания и выбрать наиболее эффективное лечение для пациента.

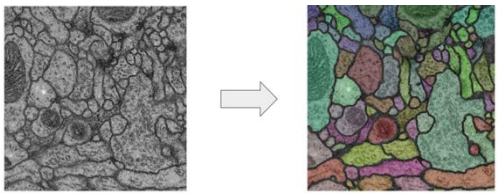


Рис. 4. Пример работы U-Net

V. Архитектура DUCK-Net

DUCK-Net является новой архитектурой сверточной нейронной сети, изначально разработанной для сегментации изображений полипов [9].

Ключевой особенностью DUCK-Net является использование блока DUCK, который использует комбинации из одного, двух и трех остаточных блоков, чтобы симулировать ядра размером 5x5, 9x9 и 13x13. Это позволяет сети использовать различные размеры ядер и компенсировать недостатки одного способа симуляции ядра над другим. Блок DUCK используется параллельно, чтобы сеть могла выбирать, какое поведение лучше подходит на каждом шаге. Это позволяет сети захватывать особенности на разных масштабах и компенсировать недостатки различных типов свертки.

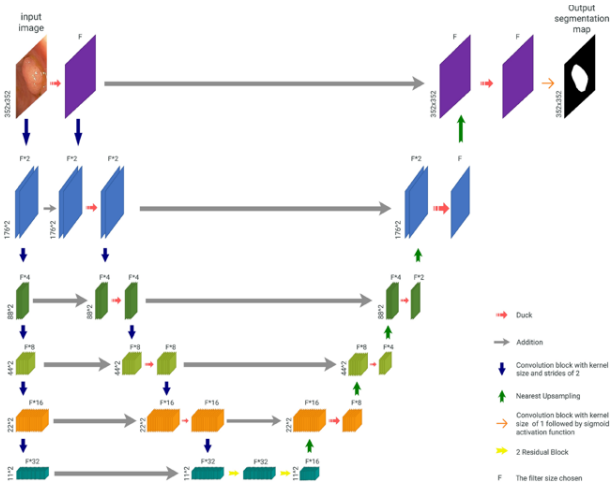


Рис. 5. Схема архитектуры DUCK-Net

Создатели архитектуры сравнили производительность блока DUCK с простым сверточным блоком в контексте архитектуры DUCK-Net, используя набор данных Kvasir-SEG, и блок DUCK последовательно превосходил простой сверточный блок по всем измеряемым метрикам производительности, как для модели с 17, так и для модели с 34 фильтрами.

По сравнению с U-Net, DUCK-Net обладает преимуществом в том, что использование большего количества блоков в DUCK позволяет захватывать признаки на разных масштабах и компенсировать недостатки различных типов сверток. Это способствует улучшению производительности модели на сложных задачах сегментации. Однако, необходимо учитывать, что увеличение числа блоков в DUCK приводит к увеличению вычислительной сложности, что может потребовать больших ресурсов, особенно при развертывании модели в ограниченных по ресурсам средах. DUCK-Net также обладает хорошей обобщающей способностью и способен достигать отличных результатов даже при ограниченных обучающих данных, что подтверждает его потенциал для использования в различных задачах сегментации.

Архитектура DUCK-Net была протестирована на нескольких наборах данных, таких как Kvasir-SEG, CVC-ClinicDB, CVC-ColonDB и ETIS-LARIBPOLYPDB, где она продемонстрировала выдающиеся результаты, превосходя многие существующие модели по метрикам Dice коэффициента, индексу Жаккара, полноте и точности. Это подтверждает её способность эффективно обрабатывать изображения с полипами различных форм, размеров и текстур. DUCK-Net использует структуру кодировщика-декодера с механизмом остаточной субдискретизации, что позволяет эффективно захватывать и обрабатывать информацию в нескольких разрешениях. Кроме того, техника увеличения данных помогает улучшить общую производительность модели, особенно при ограниченном объеме обучающих данных.

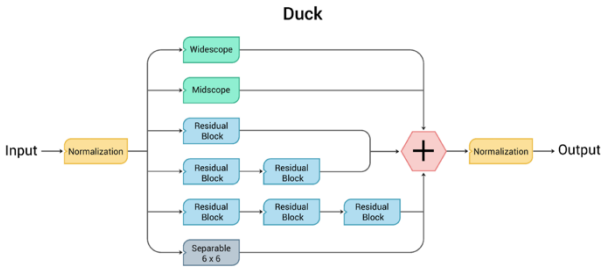


Рис. 6. Duck блок

В блоке DUCK используются остаточные блоки, впервые представленные в ResUNet++ [10], чтобы лучше понимать мелкие детали, характерные для полипов. Помимо этого, блоки Midscope и Widescope, использующие дилатационные свертки, позволяют сети распознавать более высокоуровневые признаки.

Блок Midscope симулирует ядра размером 9x9 с помощью комбинации двух остаточных блоков и последующих дилатационных сверток (рисунок 7).

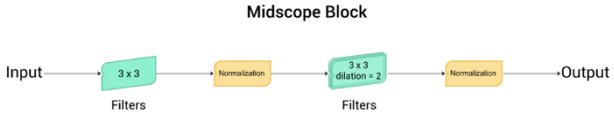


Рис. 7. Midscope блок

Блок Widescope (рисунок 8), в свою очередь, использует три остаточных блока для симуляции ядер размером 13x13. Эти блоки эффективно уменьшают количество параметров, необходимых для симуляции крупных ядер,

сохраняя при этом способность сети понимать сложные особенности изображений.

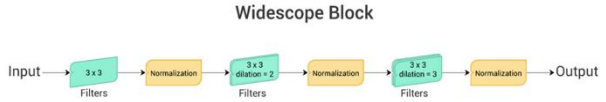


Рис. 8. Widescope блок

В ходе исследований производительности блока DUCK было показано, что он значительно превосходит простые сверточные блоки, что подтверждает его эффективность для задач сегментации изображений.

VI. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе алгоритмов глубокого обучения использовались два открытых набора данных.

A. *Lung segmentation dataset*

Данный набор [11] данных разработан консорциумом по созданию изображений с открытым исходным кодом [12] для прогнозирования тяжести снижения функции легких пациента на основе компьютерной томографии. Затем модифицирован для использования в задачах сегментации легких [13].

Датасет содержит 407 снимков КТ, которые разделены на 17011 срезов размером 512 на 512 пикселей, и масок к ним. При модификации исходные файлы nrrd были повторно сохранены в формате single tensor с масками и соответствующими меткам (легкие, сердце, трахея) в виде массивов numpy в формате pickle. В случае нашего исследования использовались исключительно маски, соответствующие легким (рисунок 9).

Каждый тензор имеет следующую форму: количество срезов, ширина, высота, количество классов, где ширина и высота количества срезов являются индивидуальными параметрами каждого идентификатора тензора, а количество классов равно 3.

Кроме того, данные были повторно сохранены в виде изображений RGB, где каждое изображение соответствует одному фрагменту идентификатора, а их изображения-маски имеют каналы, соответствующие трем классам: легкие, сердце, трахея.

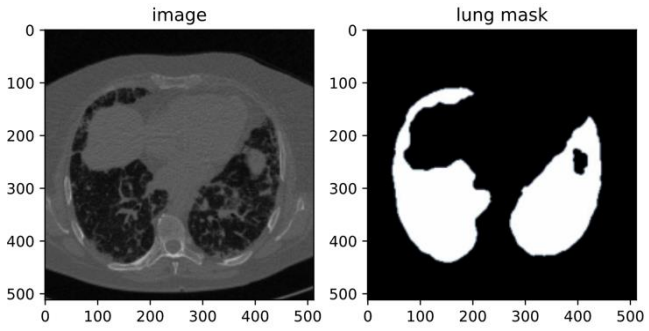


Рис. 9. Пример КТ снимка и соответствующей маски легких из Lung segmentation dataset

B. *Brain Tumor Dataset*

Данный набор данных [14] содержит 3064 пары МРТ-изображений головного мозга и соответствующих им бинарных масок, указывающих на опухоль (рисунок 10).

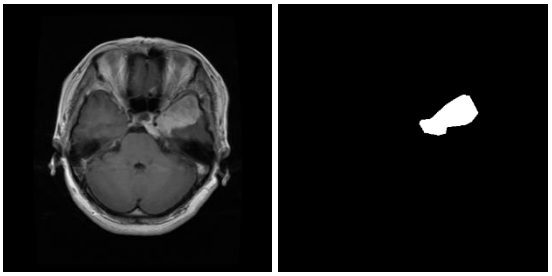


Рис. 10. Пример МРТ снимка головного мозга и соответствующей маски опухоли из Brain Tumor Dataset

VII. Функции оценки

A. *Accuracy*

Первой метрикой для задачи сегментации мед-снимков была выбрана accuracy, которая рассчитывается как отношение правильно классифицированных пикселей к общему количеству пикселей.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

B. *Mean Intersection Over Union*

Второй метрикой была выбрана MIOU (Mean Intersection over Union). Она показывает среднее значение пересечения над объединением для всех классов объектов на изображении

$$IoU = \frac{TP_c}{TP_c + FP_c + FN_c}, \quad (2)$$

$$mean\_IoU = \frac{1}{C}\sum_c IoU_c \quad (3)$$

MIOU принимает значения от 0 до 1, где 1 означает идеальное совпадение предсказанных и истинных масок сегментации. Чем выше значение MIOU, тем лучше качество сегментации.

C. *Dice коэффициент*

$$Dice\ Coefficient = \frac{2*intersection}{union+intersection} = \frac{2TP}{2TP+FN+FP} \quad (4)$$

Коэффициент Дайса (Dice coefficient) (так же называется коэффициент Сёренсена — Sorensen–Dice coefficient) или Жаккара (Jaccard similarity coefficient), который показывает меру сходства — в данном случае, показывающий меру площади правильно отмеченных сегментов (отношение площади пересечения к площади объединения).

VIII. Проведенные исследования и результаты

A. *Предобработка датасетов*

Далее приведены действия, которые были применены к обоим наборам данных “Lung segmentation dataset” и “Brain Tumor Dataset”.

Для снижения вычислительных затрат при обучении изображения были масштабированы до 352 x 352 пикселя. Из-за проблем с наложением спектров при масштабировании изображений [15] мы использовали фильтр Ланцоша [16] для сохранения качества.

Также была проведена аугментация тренировочного набора данных, что значительно улучшило обобщающие возможности модели, до такой степени, что методы регуляризации, такие как dropout, стали ненужны. Для этого была использована библиотека Albumentations [17]. Аугментация производилась перед каждой эпохой с помощью следующих техник:

1. Горизонтальные и вертикальные перевороты.
2. Изменение цвета с коэффициентом яркости из диапазона [0.6, 1.6], контрастом 0.2, насыщенностью 0.1 и цветовым тоном (hue) 0.01.
3. Аффинные преобразования с поворотами на угол, равномерно выбранный из [-180°, 180°], горизонтальными и вертикальными трансляциями на величину, равномерно выбранную из [-0.125, 0.125], масштабированием на величину, равномерно выбранную из [0.5, 1.5], и сдвигом на угол, равномерно выбранный из [-22.5°, 22°].

Наборы данных были разделены на тренировочную, тестовую и валидационную выборки в соотношении 80%–10%–10%.

Отдельно для набора данных “Lung segmentation dataset” была проведена работа с размеченными масками – были удалены разметка на трахею и сердце и оставлена только на легкие за счет подбора порога бинаризации изображения маски.

B. *Параметры обучения моделей*

Обе модели были реализованы с использованием фреймворка Tensorflow, а их обучение производилось со следующими параметрами:

1. Оптимизатор RMSprop с коэффициентом обучения (learning rate) 0,0001.
2. Разбиение данных на пакеты размером 4 изображения на протяжении 35 эпох.
3. Обучение моделей выполнялось на графическом процессоре NVIDIA Tesla P100.

C. *Результаты*

В ходе экспериментов были обучены и протестированы модели DUCK-Net и U-Net на двух наборах данных – Lung Segmentation Dataset для сегментации легких на КТ снимках и Brain Tumor Dataset для сегментации опухолей мозга на МРТ снимках.

Для модели DUCK-Net использовались две стратегии обучения:

1. Обучение с нуля (no pre-training) на целевых наборах данных Lung Segmentation Dataset и Brain Tumor Dataset.
2. Дообучение на целевых наборах данных предобученной (pre-trained) на датасете CVC-ClinicDB [18]. Датасет CVC-ClinicDB содержит видеозаписи колоноскопии и маски сегментации полипов.

В таблице 1 представлены результаты работы всех моделей, использованных в работе с первым датасетом.

ТАБЛИЦА I. Полученные результаты на датасете Lung Segmentation Dataset

Модель	***Accuracy***	***MIOU***	***Dice***
DUCK-Net (no pre-training)	0.9612	0.6632	0.7975
DUCK-Net (pre-trained)	0.9637	0.6789	0.8088
U-Net	0.7546	0.5167	0.6043

Учитывая, что обучение модели велось в условиях ограниченных вычислительных мощностей и относительно малого количества обучающих примеров, модель DUCK-Net дала хороший результат, а при обучении U-Net напротив, не удалось достичь существенных метрик на этом наборе данных.

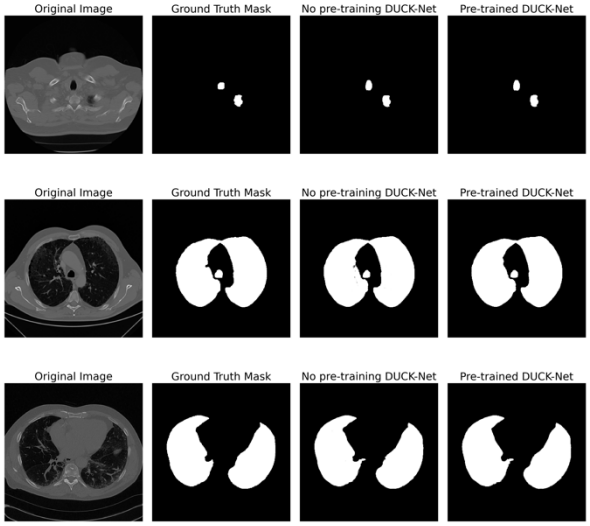


Рис. 11. Примеры сравнения эталонных масок со спрогнозированными моделями DUCK-Net (pretraining, no pretraining) на КТ снимках легких из датасета

В таблице 2 представлены результаты работы всех моделей, использованных в работе со вторым датасетом.

ТАБЛИЦА II. Полученные результаты на датасете Brain Tumor Dataset

Модель	***Accuracy***	***MIOU***	***Dice***
DUCK-Net	0.9903	0.5302	0.6930
DUCK-Net (дообучение)	0.9930	0.6274	0.7710
U-Net	0.8304	0.4802	0.6104

Модель U-Net также довольно плохо справилась с задачей на этом наборе данных, в то время как результаты дообученной DUCK-Net можно назвать приемлемыми.

Полученные результаты показывают, что архитектура DUCK-Net способна достигать высокой точности сегментации медицинских изображений даже на относительно небольших наборах данных, с которыми не справляются более старые модели вроде U-Net. Использование предобученной модели DUCK-Net и ее дообучение на целевом датасете позволяет дополнительно улучшить качество сегментации.

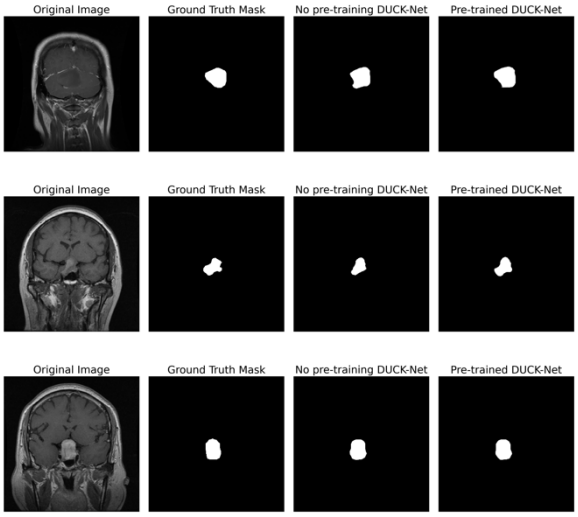


Рис. 12. Примеры сравнения эталонных масок со спрогнозированными моделями DUCK-Net (pretraining, no pretraining) на МРТ снимках из Brain Tumor Dataset

Проведенные эксперименты подтверждают перспективность использования архитектуры DUCK-Net для задач сегментации медицинских изображений, особенно в условиях ограниченного количества размеченных данных для обучения. Дальнейшие исследования могут быть направлены на проверку возможностей DUCK-Net на более широком спектре задач и датасетов медицинской визуализации.

IX. Заключение

В данной работе была рассмотрена архитектура нейронной сети DUCK-Net для сегментации медицинских изображений. Эксперименты проводились на двух наборах данных: Lung Segmentation Dataset для сегментации легких на КТ снимках и Brain Tumor Dataset для сегментации опухолей мозга на МРТ снимках. В качестве базовой модели для сравнения использовалась архитектура U-Net.

Результаты показали, что модель DUCK-Net превосходит U-Net на обоих наборах данных. На датасете Lung Segmentation Dataset DUCK-Net без предобучения достигла значений Dice коэффициента 0.7975. U-Net не справилась с задачей на этом наборе данных из-за недостаточного количества обучающих примеров, в то время как DUCK-Net показала хорошие результаты даже в условиях ограниченных данных.

На датасете Brain Tumor Dataset DUCK-Net продемонстрировала Dice 0.6930. Дообучение предобученной модели DUCK-Net позволило дополнительно улучшить метрики до значений Dice 0.7710 при тех же параметрах обучения. U-Net также не дала удовлетворительных результатов на этом наборе данных.

Таким образом, архитектура DUCK-Net показала свою эффективность и перспективность для задач сегментации медицинских изображений, особенно в условиях ограниченного количества размеченных данных для обучения. Использование предобученной модели DUCK-Net с последующим дообучением на новом целевом датасете тематики позволяет дополнительно повысить точность сегментации даже при том, что она была предварительно обучена на данных другой тематики.

Проведенное исследование подтверждает потенциал DUCK-Net как мощного инструмента для автоматизации анализа медицинских изображений. Способность модели достигать высокой точности даже на небольших наборах данных делает ее привлекательной для практического применения, учитывая сложность и затратность создания больших размеченных датасетов в медицинской сфере.

Дальнейшие исследования могут быть направлены на проверку возможностей DUCK-Net на более широком спектре задач медицинской визуализации, изучение ее робастности и обобщающей способности. Кроме того, важным направлением является повышение интерпретируемости и надежности моделей сегментации на основе глубокого обучения для их успешного внедрения в клиническую практику.

Литература

[1] What Is Lung CT Screening & How Does It Work? // emoryhealthcare.org. 2021. URL: https://www.emoryhealthcare.org/centers-programs/lung-ct-program/about.html (дата обращения 10.05.2024).

[2] Guzhva N. S. et al. 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). – IEEE, 2022. – С. 1-6.

[3] Osipov A. et al. Identification and classification of mechanical damage during continuous harvesting of root crops using computer vision methods //IEEE Access. – 2022. – Т. 10. – С. 28885-28894.

[4] Кожаринов А. С. Применение информационных технологий и искусственных нейронных сетей для прогнозирования состояний спортивных зданий и сооружений. Основные аспекты и результаты одного научно-технического проекта. – 2020.

[5] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. // arXiv.org. 2017. Дата обновления: 04.06.2017. URL: https://arxiv.org/abs/1702.05747 (дата обращения: 10.05.2024).

[6] Smolin A. et al. Reprojection-Based Numerical Measure of Robustness for CT Reconstruction Neural Network Algorithms //Mathematics. – 2022. – Т. 10. – №. 22. – С. 4210.

[7] Minaee S, Boykov YY, Porikli F, Plaza AJ, Kehtarnavaz N, Terzopoulos D. Image Segmentation Using Deep Learning: A Survey. // arXiv.org. 2020. Дата обновления: 15.11.2020. URL: https://arxiv.org/abs/2001.05566 (дата обращения: 10.05.2024).

[8] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in Proceedings of the Medical Image Computing and ComputerAssisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., October 2015.

[9] Dumitru R. G., Peteleaza D., Craciun C. Using DUCK-Net for polyp image segmentation //Scientific reports. – 2023. – Т. 13. – №. 1. – С. 9803.

[10] Jha D. et al. Resunet++: An advanced architecture for medical image segmentation //2019 IEEE international symposium on multimedia (ISM). – IEEE, 2019. – С. 225-2255.

[11] Kónya et al. CT Lung & Heart & Trachea segmentation. // Kaggle.com. 2020. URL: https://www.kaggle.com/sandorkonya/ct-lung-heart-trachea-segmentation (дата обращения: 10.05.2024).

[12] Open-Source Imaging Consortium. // OSICild.org. 2023. URL: https://www.osicild.org/ (дата обращения 10.05.2024).

[13] Chest CT Segmentation // kaggle.com. 2020. URL: https://www.kaggle.com/datasets/polomarco/chest-ct-segmentation (дата обращения 10.05.2024).

[14] Cheng J. et al. Brain Tumor Dataset. // figshare.com. 2017. URL: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427/5 (дата обращения 10.05.2024)

[15] Parmar G., Zhang R., Zhu J. Y. On aliased resizing and surprising subtleties in gan evaluation //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – С. 11410-11420.

[16] Duchon C. E. Lanczos filtering in one and two dimensions //Journal of Applied Meteorology and Climatology. – 1979. – Т. 18. – №. 8. – С. 1016-1022.

[17] Buslaev A. et al. Albumentations: fast and flexible image augmentations //Information. – 2020. – Т. 11. – №. 2. – С. 125.

[18] Bernal J. et al. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians //Computerized medical imaging and graphics. – 2015. – Т. 43. – С. 99-111.

ГРНТИ: 28.23.15
ASJC: 1702
OECD: 01.02.EP

Исследование задачи детектирования человека с помощью компьютерного зрения

Карякин А. В.
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2312716@edu.misis.ru

***Аннотация*— В данной статье рассматривается проблема детектирования человека в компьютерном зрении, которая находит применение в различных областях, таких как системы видеонаблюдения, автономное вождение, анализ поведения толпы, робототехника и умные города. С развитием глубокого обучения и нейросетевых технологий методы детектирования человека достигли значительных успехов. В статье особое внимание уделяется нейросетевой архитектуре ByteTrack, которая демонстрирует значительные улучшения в производительности трекинга за счет учета всех детекционных боксов, включая те, у которых низкие оценки достоверности. Оценка ByteTrack на наборах данных MOT17 и MOT20 показала высокую точность и эффективность, что делает её перспективной для применения в реальных задачах компьютерного зрения.**

***Ключевые слова* — детектирование человека, компьютерное зрение, глубокое обучение, нейронные сети, ByteTrack, MOT17, MOT20, мультиобъектный трекинг, системы видеонаблюдения.**

I. Введение

Задача детектирования человека в компьютерном зрении является одной из наиболее востребованных и сложных задач, находящих свое применение в различных областях, таких как системы видеонаблюдения, автономное вождение, анализ поведения толпы, робототехника и умные города. Эффективное детектирование человека позволяет обеспечивать безопасность, автоматизировать процессы и улучшать взаимодействие между человеком и машиной.

С развитием глубокого обучения и нейросетевых технологий методы детектирования человека достигли значительных успехов. Нейронные сети уже активно используются для оценки точности позиционирования трамваев в городских условиях, что значительно улучшает навигационные системы общественного транспорта [1]. Важно отметить, что нейронные сети также применяются для прогнозирования поведения транспортных средств, что имеет большое значение для автономных транспортных систем [2]. Подобные технологии также находят применение в разработке алгоритмов для беспилотных летательных аппаратов, обеспечивая высокую точность и безопасность полетов [3][4]. В медицине нейронные сети используются для улучшения качества компьютерной томографии, что способствует более точной диагностике заболеваний [5].

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. Тем не менее, несмотря на достижения в этой области, остаются нерешенными задачи такие как трекинг множества объектов в реальном времени, обработка сцен с высокой плотностью объектов и условиями окклюзии, а также минимизация ложных срабатываний и пропущенных детекций. Искусственный интеллект обладает значительным потенциалом в научных исследованиях, так как позволяет анализировать скрытые от человека закономерности и получать новые знания [6].

Традиционные подходы к детектированию человека часто сталкиваются с проблемами масштабируемости и скорости обработки данных, что делает их неприменимыми для задач реального времени. В последние годы были предложены новые нейросетевые архитектуры, которые направлены на решение этих проблем и улучшение общей производительности систем детектирования и трекинга.

В данной статье рассматривается перспективная нейросетевая архитектура ByteTrack [7]. ByteTrack отличается своей способностью учитывать все детекционные боксы, включая те, у которых низкие оценки достоверности, что позволяет минимизировать потери информации об объектах и улучшить точность трекинга. Эта архитектура демонстрирует значительные улучшения в производительности детектирования и трекинга человека, что делает её перспективной для применения в реальных задачах компьютерного зрения.

II. Наборы данных

A. MOT17

MOT17 (Multiple Object Tracking 17) является одним из самых широко используемых наборов данных для мультиобъектного трекинга [8]. Он состоит из видеозаписей семи различных сцен, снятых как в помещениях, так и на улице, с участием пешеходов.

В MOT17 используются три различных детектора для создания аннотаций: SDP, Faster R-CNN и DPM, что позволяет моделям обучаться на разных вариантах детекций. Это помогает обеспечить объективное сравнение различных методов трекинга. Набор данных поддерживает как онлайн, так и оффлайн подходы к

трекингу, где оффлайн методы могут использовать будущие кадры для улучшения предсказаний. Общая цель MOT17 заключается в оценке способности алгоритмов точно определять и отслеживать объекты в сложных и динамичных условиях.

На рисунке 1 отображены кадры из различных клипов с данного датасета, которые включают в себя различные сцены:

1) снятые из автобуса на оживленном перекрёстке (а);
2) снятые на переполненной пешеходной улице стационарной камерой (б);
3) кадры пешеходной улицы ночью со смотровой площадки (в);
4) снятые с движущейся камерой в оживлённом торговом центре (г).

а

б

в

г

Рис. 1. Примеры кадров MOT17

B. MOT20

MOT20 (Multiple Object Tracking 20) отличается высокой плотностью объектов и сложными условиями окклюзии [9]. Этот набор данных представляет 8 сцен (4 для обучения и 4 для тестирования) с большим количеством пешеходов, что создает значительные вызовы для алгоритмов трекинга. В среднем, в кадре присутствует около 170 пешеходов, что делает этот датасет критически важным для тестирования алгоритмов в экстремальных условиях.

MOT20 включает видео, снятые в различных условиях, таких как улицы города, торговые центры и общественные мероприятия, что позволяет моделям обучаться на широком спектре сцен. Этот набор данных предназначен для оценки производительности моделей в условиях высокой плотности объектов и частых окклюзий, что особенно важно для применения в реальных системах видеонаблюдения и анализа поведения.

На рисунке 2 отображены кадры из различных клипов с данного датасета, которые включают в себя различные сцены:

1) кадры переполненной площади в ночное время (а);
2) кадры людей, покидающих стадион в ночное время (б);
3) кадры с переполненного крытого вокзала (в).

а

б

в

Рис. 2. Примеры кадров MOT20

III. Анализ ByteTrack

Архитектура ByteTrack является новаторским подходом к задаче мультиобъектного трекинга (MOT). Основной целью MOT является определение границ и идентичностей объектов в видео. ByteTrack отличается от традиционных методов тем, что включает в процесс ассоциации практически все детекционные боксы, а не только те, у которых высокие оценки достоверности. Это позволяет уменьшить количество пропущенных истинных объектов.

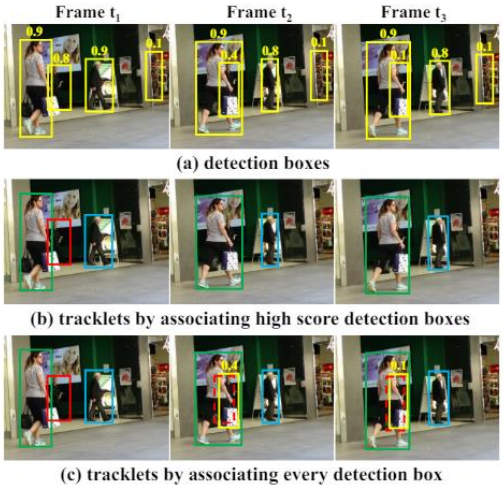


Рис. 3. Примеры кадров при дневном и ночном освещении

Основная идея ByteTrack заключается в использовании метода ассоциации, который учитывает все детекционные боксы, включая боксы с низкими оценками. Эти боксы, как правило, игнорируются другими методами, что приводит к потере информации об объектах, которые могут быть частично закрытыми или находиться в сложных условиях. В ByteTrack для боксов с низкими оценками используется их схожесть с треклетами (маленькими треками), чтобы восстановить истинные объекты и отфильтровать детекции фона.

Архитектура ByteTrack включает два основных этапа ассоциации. Сначала высоко оцененные детекционные боксы связываются с треклетами на основе схожести движений или внешнего вида, используя фильтр Калмана для предсказания местоположения треклетов в новом кадре. Сходство вычисляется по коэффициенту пересечения (IoU) или дистанции по признакам Re-ID (идентификации повторных появлений). Затем проводится второй этап ассоциации, на котором оставшиеся несоответствующие треклеты связываются с низко оцененными детекционными боксами на основе схожести движений. Это позволяет корректно восстановить объекты, которые были частично закрыты, и устранить детекции фона.

ByteTrack также был интегрирован с YOLOX [10], современным детектором объектов, для создания мощного трекера, который показал высокую производительность на различных тестовых наборах.

IV. Метрики

В рамках данного исследования была протестирована работа предобученных моеделей ByteTrack на датасетах MOT17 и MOT20.

Результат работы оценивался по следующим показателям:

1. Precision - показывает, сколько из всех обнаруженных объектов действительно являются людьми

2. Recall - показывает, сколько из всех людей, которые должны были быть найдены, модель действительно нашла.

3. MOTA - это общая оценка работы модели по отслеживанию людей. Она учитывает, сколько людей модель пропустила, сколько лишних объектов нашла и сколько раз перепутала людей между собой.

4. MOTP – показывает, насколько точно модель определяет местоположение людей. Она измеряет, насколько близко предсказанные позиции людей к их настоящим позициям

5. IDF1 - это комбинированная оценка, которая учитывает как точность, так и полноту в распознавании и отслеживании конкретных людей.

6. IDs - показывает, сколько раз модель перепутала людей между собой, то есть присвоила одному человеку идентификатор другого.

Предобученная на датасетах CrowdHuman, MOT17, Cityperson и ETHZ модель ByteTrack показала следующие результаты на MOT17 (обучающая часть):

ТАБЛИЦА I. Оценка на MOT17

	bytetrack_x_mot17
Precision	0.978
Recall	0.925
MOTA	0.9
MOTP	0.122
IDF1	0.832
IDs	426

Предобученная на датасетах CrowdHuman и MOT20 модель ByteTrack показала следующие результаты на MOT20 (обучающая часть):

ТАБЛИЦА II. Оценка на MOT20

	bytetrack_x_mot20
Precision	0.983
Recall	0.951
MOTA	0.934
MOTP	0.138
IDF1	0.893
IDs	1049

Модели показали хорошие показатели, близкие к результатам, полученными авторами. Это показывает высокое качество моделей.

Также в качестве испытания на вход нейросети была подана видеозапись с камеры видеонаблюдения г. Гусь-Хрустальный (рисунок 4). Модель продемонстрировала хорошие результаты, корректно детектируя и отслеживая людей даже в небольшом масштабе на дальних дистанциях.

Рис. 4. Пример работы ByteTrack

V. ЗАКЛЮЧЕНИЕ

В данной статье была рассмотрена нейросетевая архитектура ByteTrack для детектирования человека, которая отличается новаторским подходом к ассоциации всех детекционных боксов, включая те, у которых низкие оценки достоверности.

Это позволило минимизировать потери информации и значительно улучшить точность трекинга. ByteTrack использует продвинутые методы ассоциации и интегрирован с современными детекторами объектов, что обеспечивает высокую производительность и точность.

Проведенные эксперименты на наборах данных MOT17 и MOT20 показали, что ByteTrack является очень эффективной моделью опираясь на ключевые метрики, таким как Precision, Recall, MOTA, MOTP и IDF1. Эти результаты подтверждают высокую эффективность ByteTrack, и доказывают показатели, полученные авторами.

Таким образом, ByteTrack представляет собой перспективное решение для задач компьютерного зрения в реальных условиях, таких как системы видеонаблюдения и автономное вождение. Будущие исследования могут быть направлены на дальнейшее улучшение точности и скорости обработки, а также адаптацию модели к новым сценариям и условиям эксплуатации.

ЛИТЕРАТУРА

[1] 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

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

[4] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[5] Smolin A, Yamaev A, Ingacheva A, Shevtsova T, Polevoy D, Chukalina M, Nikolaev D, Arlazarov V. Reprojection-Based Numerical Measure of Robustness for CT Reconstruction Neural Network Algorithms. Mathematics. 2022; 10(22):4210. https://doi.org/10.3390/math10224210 (Accessed: December 26, 2023).

[6] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-3-93-105.

[7] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, Xinggang Wang, “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”, arXiv, 2021, 2110.06864

[8] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, Konrad Schindler, “MOT16: A Benchmark for Multi-Object Tracking”, arXiv, 2016, 1603.00831

[9] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, Laura Leal-Taixé, “MOT20: A benchmark for multi object tracking in crowded scenes”, arXiv, 2020, 2003.09003

[10] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun, “YOLOX: Exceeding YOLO Series in 2021”, arXiv, 2021, 2107.08430

[11] “ByteTrack Github repository”, available at: https://github.com/ifzhang/ByteTrack, (Acessed: May 20)

Исследование возможности детектирования трещин и дорожных заплаток на асфальте

В. О. Кирвяков
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1809131@edu.misis.ru

***Аннотация*— В данном исследовании анализируются решения, основанные на коммерческом программном обеспечении, и проведено сравнение возможностей распознавания трещин и дорожных заплаток на асфальте, в разных погодных условиях, времени суток, а также перекрытий и искажений. Работа выполнена на данных, полученных с камеры передвижной лаборатории, предоставленных компанией НПО Регион, а также дополненных собственными снимками. В работе использовались одни из самых распространённых в области детекции и классификации изображений нейронные сети – Yolo и VGG.**

Ключевые слова — компьютерное зрение, детекция, трещин на асфальте, распознавание дорожных заплаток на асфальте, YOLO, VGG

I. Введение

Автоматическое обнаружение дефектов на дорожном покрытии является ключевым аспектом технического обслуживания транспорта для обеспечения безопасности движения. Тем не менее, это остается сложной задачей из-за разнообразия и интенсивности повреждений, а также сложности фона на изображениях, например, слабого контраста покрытия и возможных теней. Недавние успехи в применении методов глубокого обучения компьютерного зрения поспособствовали созданию методов обнаружения дефектов с использованием сверточных нейронных сетей.

В работе [1] представлена интегрированная система для обнаружения и определения признаков дефектов дорожного покрытия, в работе [2] – полный набор алгоритмов обработки изображений. Существующие методы обработки недостаточны для различения дефектов и сложного фона на изображениях среднего или низкого качества. Эффективность в этой области была продемонстрирована глубокими нейронными сетями, отличающимися высокой производительностью в решении многих задач. Глубокие сверточные нейронные сети особенно популярны для обучения с учителем. Обнадеживающие результаты этих моделей являются основной мотивацией для применения методов глубокого обучения в задачах обнаружения дефектов дорожного покрытия.

В контексте развития технологий глубокого обучения [3] проблема выбора оптимальных архитектур нейронных сетей для задачи детектирования дорожных дефектов приобретает особую актуальность. Среди наиболее распространенных архитектур выделяются YOLO и VGG, которые представляют собой различные подходы к решению задач компьютерного зрения [4, 5].

Архитектура YOLO [6] отличается от других методов тем, что осуществляет детекцию объектов на изображении за один проход, анализируя его целиком. Главной особенностью данной сети является её способность точно идентифицировать множество различных объектов на сложных и разреженных изображениях дорожных сцен.

В свою очередь, архитектура VGG [7] привлекает внимание своим глубоким строением и серией сверточных слоев. Этот подход обеспечивает высокую точность классификации, но может потребовать больше вычислительных ресурсов в сравнении с YOLO [8].

Исследование направлено на сравнение эффективности нейронных сетей YOLO и VGG в контексте детектирования трещин и дорожных заплаток на асфальте. Анализ и сравнение этих двух архитектур позволят выявить их преимущества и ограничения в контексте данной задачи.

II. Наборы данных

Для проведения процессов обучения и тестирования нейронных сетей, рассматриваемых в данном исследовании, были использованы два различных набора данных. Эти наборы включают как локально собранные авторами данные, так и ограниченные для открытого доступа данные, предоставленные компанией НПО Регион. Рассмотрим эти наборы более детально. Для их создания были применены современные методы сбора информации [9].

A. Коммерческий (закрытый) набор данных

Обширный архив данных, предоставленный организацией, включает в себя видеозаписи дорожных сценариев, зафиксированных на дорогах регионов России. Набор данных содержит аннотированные видеопоследовательности, записанные при различных условиях освещения и погодных условиях с использованием мобильной лаборатории [10], представленной на рисунке 1. Кроме того, набор данных содержит различные типы дорожных дефектов, из которых в исследовании рассматриваются три класса, представленные на рисунке 2: *трещина малая, трещина крупная, дорожная заплатка.*

Рисунок 1. Передвижная лаборатория НПО Регион

B. Собственный набор данных

В данной статье так же используется локальный набор данных, собранный автором, состоящий из фотографий знаков в ночное время суток. Локальный набор данных был использован по причине отсутствия в датсете НПО Регион примеров дорожных знаков в ночное время, ввиду не надобности подобных снимков для работы компании.

Данные получены с видеорегистраторов и фотокамеры смартфона и представлены на рисунке 2.

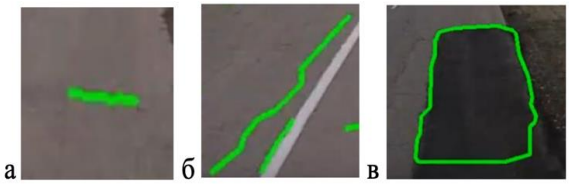


Рисунок 2. Примеры классов дорожных дефектов: а) малая трещина, б) крупная трещина, в) дорожная заплатка

На рисунке 3 отображены сложные для распознавания ситуации:

- Пролитые трещины, схожи с заплатками;
- Искаженная форма дорожных заплаток;
- Большое количество трещин в одном кадре;
- Разломы асфальта;
- Перспективные искажения;
- Засветы и блики на дорогах из-за погодных условий;

Рисунок 3. Примеры сложных ситуаций

A. YOLOv3

В исследовании [11] рассматривается задача распознавания дорожных дефектов. Предложено использование детектирующей нейронной сети и классификация дефектов по категориям. В работе применяется архитектура YOLOv3, представленная на рисунке 8. Модель обучена для обнаружения и распознавания дорожных дефектов, при этом выходной слой нейронной сети включает сорок два класса, охватывающих различные виды дефектов дорожного покрытия. Архитектура нейронной сети была адаптирована для учета всех необходимых классов. Пример детектирования отображен на рисунке 4.

Для обучения нейронной сети использовался датасет, предоставленный НПО "Регион". Обучение проводилось на примерно 6852 кадрах с размеченными дорожными дефектами, а тестирование – на отдельном датасете, состоящем из примерно 700 кадров с 2341 дефектом. В качестве критерия оценки использовалась функция потерь mAP (средняя площадь под кривой точность–полнота).

Модель YOLOv3 обучалась на 1500 батчах, каждый из которых содержал 64 изображения, с постоянной скоростью обучения. Размер изображений составлял 608x608 пикселей, что являлось компромиссом между скоростью и качеством. В процессе обучения также применялись методы аугментации, такие как изменение оттенка, насыщенности, экспозиции, а также батч-нормализация. Кроме того, каждые 10 батчей разрешение изображений менялось с 608x608 на разрешения, кратные 32, для повышения устойчивости модели к различным масштабам. Также была проведена работа по балансировке обучающей выборки. [12].

Рисунок 4. Пример работы YOLOv3 в задаче детектирования трещин и дорожных заплаток на асфальте

B. VGG

Архитектура VGG [13] представлена на рисунке 9, часто используется для задач классификации изображений, однако может быть адаптирована и для задачи детектирования объектов, таких как дорожные дефекты. Пример детектирования отображен на рисунке 5.

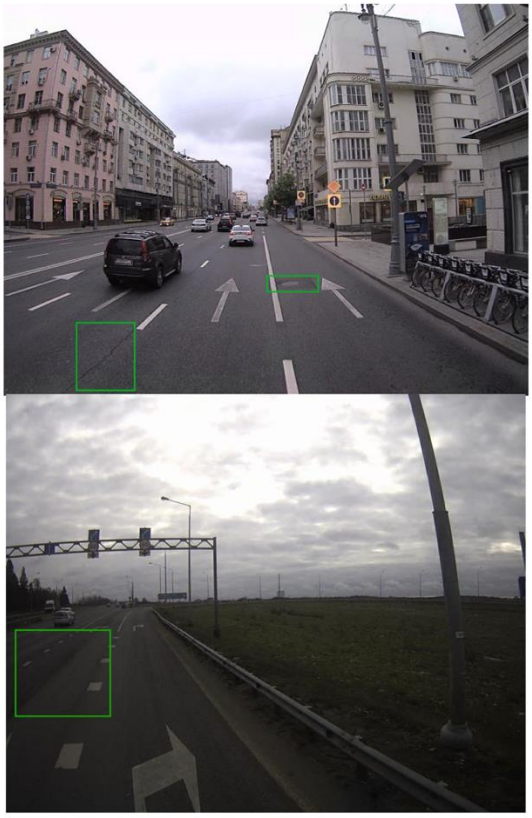


Рисунок 5. Пример работы VGG в задаче детектирования трещин и дорожных заплаток на асфальте

Архитектура VGG [14] для детектирования дорожных дефектов может включать следующие блоки:

1. Входной слой: подача изображения дорожного участка или кадра на вход нейронной сети.

2. Сверточные слои: использование нескольких последовательных сверточных слоев с небольшими ядрами (обычно 3x3) для извлечения различных признаков из входного изображения. Эти слои помогают выделять узоры и характеристики изображения.

3. Подвыборка: возможность после каждого сверточного слоя использования также слоя подвыборки, такого как слой субдискретизации (max-pooling), уменьшающего размер признаковых карт, сохраняя наиболее важные признаки.

4. Полносвязные слои: передача выходов последнего сверточного слоя через один или несколько полносвязных слоев, используемых для классификации. В контексте детектирования объектов эти слои могут быть адаптированы для выдачи более детальных предсказаний о местоположении объектов.

5. Выходной слой: генерирование выходным слоем предсказаний, включая вероятности присутствия различных классов дорожных дефектов и информацию об их местоположении (ограничивающие прямоугольники).

6. Функция потерь: использование в ходе обучения сети функции потерь, оценивающей разницу между предсказанными значениями и истинными метками. В задаче детектирования может включать в себя компоненты, связанные с классификацией и локализацией объектов.

7. Обучение и дообучение: обучение сети на размеченном наборе данных дорожных дефектов. При необходимости сеть может быть дообучена на специфическом наборе данных для улучшения ее способности обнаружения конкретных дорожных дефектов.

Архитектура VGG обеспечивает глубокое извлечение признаков, что может быть полезным для выделения характерных черт дорожных дефектов при их детектировании. VGG обучался на том же датасете, что и вышеприведенный YOLOv3.

III. Сравнение

Сравним два описанных подхода. Для сравнения используется отдельно собранный набор данных – 700 изображений с 2341 размеченным дорожным дефектом. Качество работы двух подходов складывается из качества работы локализующей и классифицирующей частей. Оценка локализации производится при помощи расчёта меры Жаккара (Intersection over Union, IoU) для каждой детекции.

Введём следующие величины:

Разработка конечной системы велась в две стадии:

- TP – детектор верно локализовал дорожный дефект и определил его класс.
- FP – детектор нашёл дорожный дефект там, где его нет, или не верно определил его класс.
- FN – детектор не нашёл дорожный дефект, хотя он есть и для него есть разметка.

Стоит отметить, что TN в данном случае не определена, так как эта величина означает отсутствие определения дорожного дефекта, где его действительно нет. По введённым величинам строятся такие функции оценок, как:

- $Precision = \frac{TP}{TP+FP}$ – количество обнаружений детектом дорожного дефекта, где он действительно есть, по отношению к общему числу предсказанных объектов;
- $Recall = \frac{TP}{TP+FN}$ – количество обнаружений детектом дорожного дефекта из действительно присутствующих в кадрах;
- $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$ – оценка баланса между точностью (precision) и полнотой (recall).

Таблица 1 отображает количественные оценки для двух подходов.

ТАБЛИЦА I. Оценка ошибок работы детекторов

	VGG	YOLO v3	VGG	YOLO v3
TP	2232	2094	478	432
FP	31	132	50	64
FN	78	115	51	83
Precision	0,98	0,94	0,9	0,87
Recall	0,96	0,95	0,9	0,83
F1	0,97	0,94	0,9	0,85

Исходя из данных таблицы, детекторы VGG и YOLOv3 имеют незначительные отличия, что означает их успешное определение действительных дорожных дефектов. В данную таблицу включены оценки тестирования сетей на двух датасетах, а то, что оценки, полученные на разных наборах данных несильно отличаются, показывает устойчивость нейронных сетей.

В оценку классифицирующей части включены все объекты, которые входят во множество TP локализующей части. Классифицирующая нейронная сеть YOLOv3 на выходе имеет 3 класса, так же, как и классификатор VGG, в связи с этим матрицы ошибок имеют одинаковые размеры. На рисунке 6 отображена матрица ошибок и отчёт о классификации, содержащий precision, recall и F1-меру [15, 16], для нейросети VGG. Здесь номера классов 1–3 означают классы дорожных дефектов: *трещина малая, трещина крупная, дорожная заплатка.*

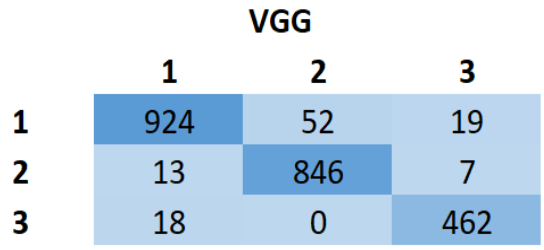


Рисунок 6. Матрица соответствий ошибок VGG

Таким образом, все классы распознаются с достаточно высоким значением F1-меры и при этом распознались все классы. Равномерное распределение метрик качества можно объяснить хорошо составленной выборкой. Что касается точности детектирования и распознавания, по этим результатам видно, что данная архитектура нейронной сети подходит к нашей задаче.

Классифицирующая часть YOLOv3 на выходе имеет так же 3 класса. Поэтому можно определить некоторую переходную матрицу ошибок, которая содержала бы 3 реальных класса. Данная матрица представлена на рисунке 7.

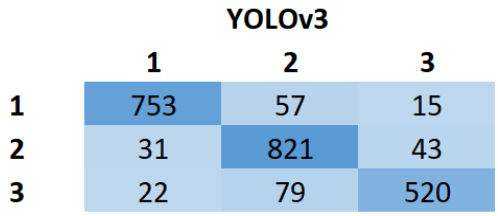


Рисунок 7. Матрица соответствий ошибок YOLOv3

Численные оценки классификации этой нейросети имеют значения ниже предыдущих. Такую несущественную с предыдущей нейронной сетью разницу в качестве можно объяснить несколькими обстоятельствами: устойчивость YOLOv3 к таким искажениям как: перепады света, перспективные искажения, большее число тестовых картинок, так как более качественный детектор смог правильно локализовать дорожные дефекты, которые затем и классифицировались. Однако детектор VGG в ситуациях с множественными объектами, показал лучший результат.

Сравнивая классификаторы YOLOv3 и VGG, можно заметить, что все классы, представленные в собранном датасете распределены равномерно. VGG распознаёт эти классы лучше из-за упомянутых выше обстоятельств. Сбалансированность распознанных классов ещё раз показывает важность составления широкой репрезентативной обучающей выборки.

IV. Заключение

В данной работе были рассмотрены два набора данных, на которых обучались и тестировались рассматриваемые нейронные сети. Приведены два подхода – локализации и классификации к детектированию дорожных дефектов: модернизация YOLOv3, предложенная и обученная авторами работы [11], в которой решалась задача детектирования дорожных дефектов, и VGG [14], обученная на том же наборе данных. Каждая сеть рассмотрена с точки зрения её архитектуры, процесса обучения, используемых для обучения и тестирования наборов данных.

Приведённые подходы были сравнены на собственном наборе данных и датасете, предоставленном компанией НПО Регион. Отдельно были оценены качество локализации и классификации дефектов. Данное исследование показало, что обе нейронных сети справились с задачей детектирования дорожных дефектов и могут применяться в задачах, связанных с этим. В зависимости от набора данных полученные результаты могут отличаться от приведенных в статье.

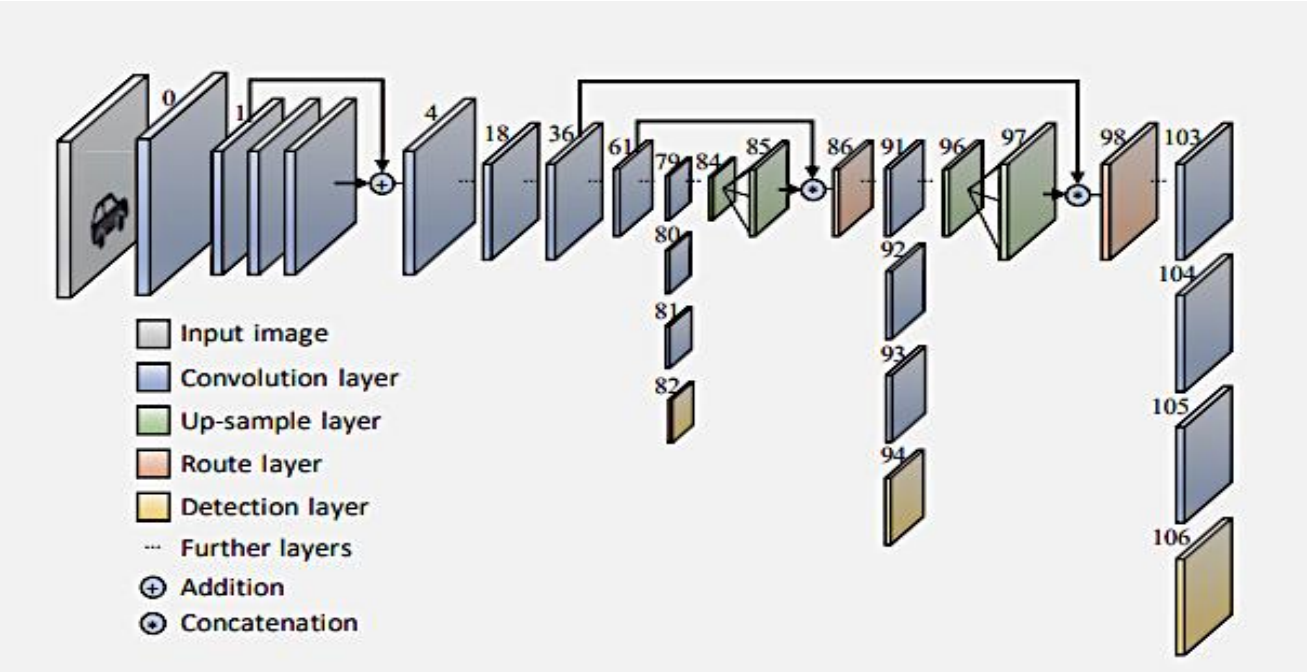


Рисунок 8. Архитектура YOLOv3

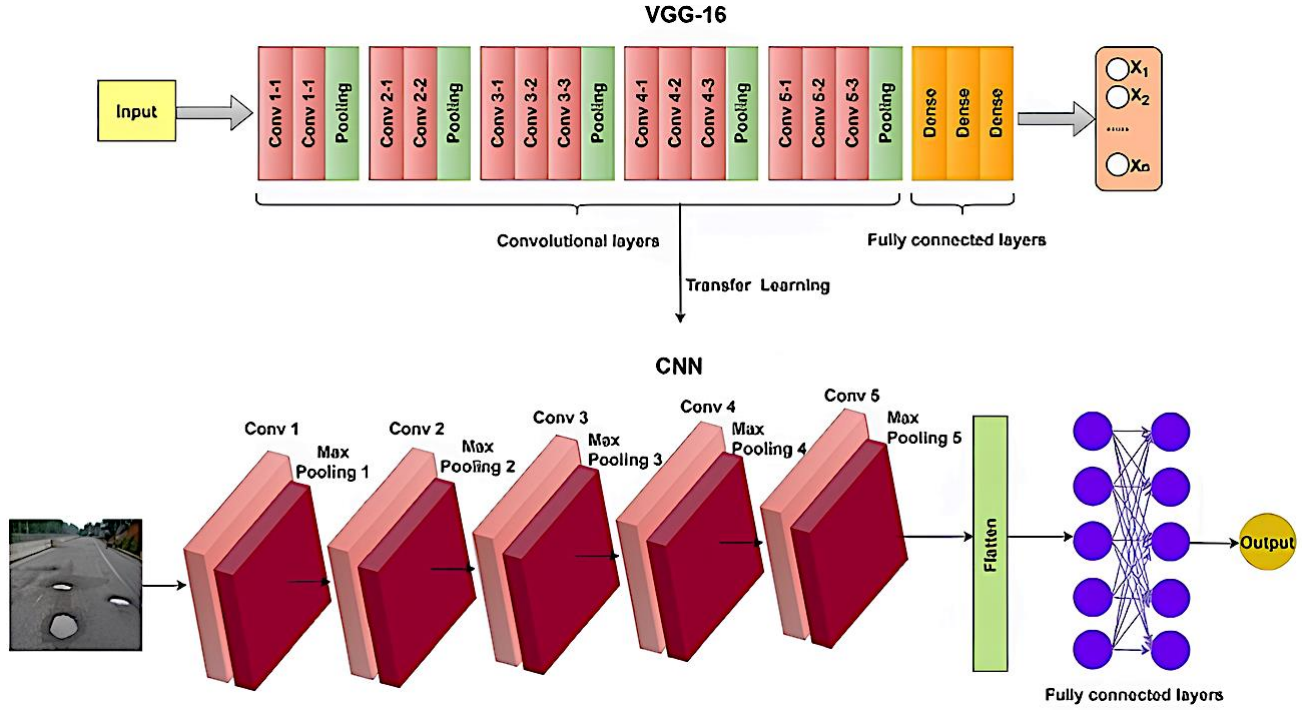


Рисунок 9. Архитектура VGG

ЛИТЕРАТУРА

[1] Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, Deep learning-based road damage detection and classification for multiple countries, Automation in Construction, vol. 132, 2021.

[2] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. pp. 21–37.

[3] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95. 10.21146/0042-8744-2022-3-93-105.

[4] Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // CoRR. — 2015 — Vol. abs/1512.03385. — 1512.03385.

[5] 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.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, Yolov4:Optimal speed and accuracy of object detection, arXivpreprint arXiv:2004.10934, 2020.

[7] Pawe Staszewski, Maciej Jaworski, Jinde Cao, Fellow, IEEE and Leszek Rutkowski, Fellow “A new

approach to descriptors generation for image retrieval by analyzing activations of deep neural network layers", IEEE.

[8] Hoang, Lee, "An Evaluation of VGG16 and YOLO v3 on Hand-drawn Images" (2019). University Honors Theses. Paper 693

[9] R. R. Bikmaev, A. A. Polukarov and R. N. Sadekov, "Visual Localization of a Ground Vehicle Using a Monocamera and Geodesic-Bound 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.

[10] НПО Регион. Режим доступа-URL: https://nporegion.ru/laboratorii/

[11] A. A. Shaghouri, R. Alkhatib, and S. Berjaoui, "Real-time pothole detection using deep learning," 2021, https://arxiv.org/abs/2107.06356.

[12] Д.Е, Иванов & Полевой, Дмитрий & Sholomov, Dmitry. (2018). Отбор информативных элементов для обучения легкого сверточного нейросетевого классификатора в условиях сильного дисбаланса обучающей выборки. 199-204. 10.14357/20790279180523.

[13] Ярышев С.Н., Рыжова В.А., Технологии глубокого обучения и нейронных сетей в задачах видеоанализа – СПб: Университет ИТМО, 2022 – 82 с.

[14] Swain, S. Automatic detection of potholes using VGG-16 pre-trained network and Convolutional Neural Network / S. Swain, A. K. Tripathy – Vellore: VIT, School of Computer Science Engineering and Information Systems, 2024. – 16 p. – ISSN 2405-8440.Possatti, Lucas C. et al. "Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars", 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8.

[15] Possatti, Lucas C. et al. "Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars", 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1-8.

[16] J. Redmon, A. Farhadi. "YOLOv3: An Incremental Improvement", ArXiv abs/1804.02767, 2018,

Исследование возможности распознания больных растений при помощи компьютерного зрения

Я. О. Кудинов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2100108@edu.misis.ru

***Аннотация*— Традиционные методы диагностики болезней растений требуют значительных временных и трудовых ресурсов. В данной работе исследуется возможность использования методов компьютерного зрения для автоматизации этого процесса. Для этой цели обучены и сравнены две модели глубокого обучения: VGG19 и InceptionV3, которые могут качественно распознавать заболевания благодаря анализу изображений листьев растений. Обе модели обучались на PlantaeK наборе данных, содержащем изображения здоровых и больных листьев восьми различных видов растений. Целью исследования является сравнительный анализ точности и эффективности распознавания заболеваний растений с использованием указанных моделей. В ходе работы проведен инференс на собственных изображениях растений.**

Ключевые слова — компьютерное зрение, глубокое обучение, растения, классификация растений, VGG19, Inceptionv3

I. Введение

Современное сельское хозяйство сталкивается с множеством вызовов, среди которых своевременное и точное выявление заболеваний растений. Болезни растений могут быстро распространяться, приводя к значительным потерям урожая и снижению качества продукции. Традиционные методы диагностики, такие как визуальный осмотр и лабораторные тесты, требуют значительных временных и трудовых ресурсов, а также наличия высококвалифицированных специалистов. В условиях крупномасштабного производства это становится трудновыполнимой задачей, что хорошо характеризует необходимость разработки автоматизированных систем для мониторинга состояния растений [1].

В последние годы значительный прогресс в области искусственного интеллекта и, в частности, глубокого обучения, открыл новые возможности для автоматизации процесса диагностики заболеваний растений. Компьютерное зрение, опирающееся на методы глубокого обучения, позволяет создавать модели, способные анализировать изображения листьев растений и с высокой точностью распознавать наличие различных заболеваний. Это не только ускоряет процесс диагностики, но и снижает вероятность человеческих ошибок [2].

Распознавание больных растений с помощью компьютерного зрения требует использование изображений листьев для идентификации признаков заболеваний. Данный подход не только экономит время и ресурсы, но и предоставляет возможность постоянного мониторинга состояния посевов, что особенно важно для крупных агропромышленных комплексов.

Использование искусственных нейронных сетей широко распространено, нейронные сети находят применение в большом спектре задач, таких как системы навигации [3], компьютерное зрение[4], автопилоты [5,6,7] и другие области [8,9]. Благодаря высокой производительности нейронных сетей существует множество моделей подходящих для анализа и классификации изображений растений. Для того чтобы успешно распознавать болезни растений, необходимо обучить модель на большом наборе данных, содержащем изображения листьев с различными заболеваниями. В результате получается эффективный инструмент для автоматизированной диагностики, который может значительно повысить точность и скорость выявления болезней [10].

Таким образом, внедрение методов глубокого обучения в процесс диагностики болезней растений может существенно повысить точность и оперативность обнаружения патогенов, что, в свою очередь, способствует улучшению управления сельскохозяйственными ресурсами и снижению потерь урожая. Целью исследования является использование и сравнительный анализ моделей VGG19 и InceptionV3 для распознавания заболеваний растений. В статье проведена оценка производительности на основе точности распознавания заболеваний на изображениях листьев.

II. Наборы данных

В данной работе для обучения данных был использован датасет plantaeK состоящий из более чем 2000 изображений здоровых и больных листьев различных растений, перед запуском нейронных сетей данный датасет разделен на тренировочную и тестовую выборки.

A. *Plantaek*

Исследуемые растения являются аборигенными для Кашмирского региона Индии, который характеризуется прохладным климатом на протяжении нескольких месяцев в году и приятной погодой в остальное время. Для исследования выбраны восемь различных растений: яблоня, абрикос, вишня, клюква, виноград, персик, груша и грецкий орех. Эти растения были отобраны на основе их коммерческой и медицинской ценности. Основное внимание уделяется листьям, так как они появляются намного раньше плодов и других частей

растений. Для каждого вида растения были собраны два типа листьев: здоровые и больные. Учитывая природные условия, в которых работают фермеры и агрономы, все изображения были сделаны днем в автоматическом режиме с помощью цифровой зеркальной камеры Nikon с параметрами ISO 100, диафрагмой F/5.6, без вспышки и выдержкой 1/640 секунды. Все фотографии сделаны с использованием объектива 18–55 мм[11]. .

а 

б 

в 

г 

Рис.1 – примеры листьев из набора данных

В. *Набор данных для проверки работоспособности*

Помимо ряда изображений из plantaek сгруппированных в тестовую выборку, позже был использован ряд собственных изображений для формирования второго тестового набора данных, среди которых было несколько изображений цветов.

а 

б 

в

Рис.2 – примеры растений из собственного набора данных

III. Нейросетевые архитектуры

A. VGG19

VGG19 (Visual Geometry Group 19) — это сверточная нейронная сеть, разработанная группой исследователей из Оксфордского университета. Она получила широкую известность после успешного участия в конкурсе ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 года, где продемонстрировала выдающиеся результаты в задачах классификации и локализации изображений. Концепция модели VGG19 такая же, как и у VGG16, за исключением того, что она имеет 19 слоев. 16 и 19 обозначают количество весовых слоев в модели (сверточных слоев) Архитектура VGG19 состоит из 19 слоев, включая 16 сверточных и 3 полносвязных слоя. Ключевой особенностью этой сети является использование очень маленьких (3x3) сверточных фильтров, что позволяет захватывать больше деталей и повышать точность распознавания.

Далее представлена полная архитектура модели VGG19:

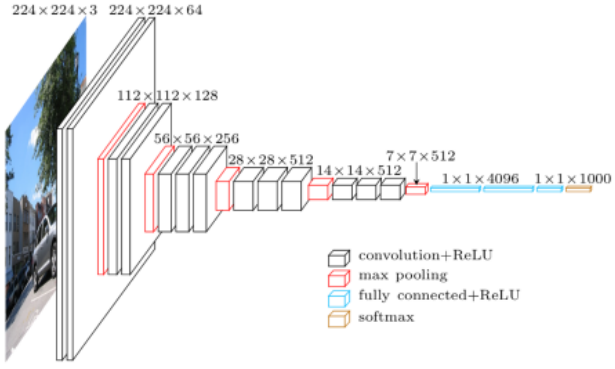


Рис.3 – архитектура сети VGG19

Входной слой(Input Layer): VGGNet принимает входное изображение размером 224×224. Для конкурса ImageNet создатели модели обрезали центральный участок размером 224×224 в каждом изображении, чтобы сохранить единообразный входной размер изображения.

Сверточные слои: VGG19 имеет 16 сверточных слоев, сгруппированных в пять блоков. Каждый блок завершается слоем подвыборки (max-pooling) с размером окна 2x2 и шагом 2. Все сверточные слои используют фильтры размером 3x3 с шагом 1 и заполнением (padding) 1, что позволяет сохранять пространственные размеры входных данных. Далее следует модуль ReLU — сокращающее время обучения. ReLU означает функцию активации выпрямленной линейной единицы; это кусочно-линейная функция, которая выводит входные данные, если они положительные; в противном случае выход равен нулю. Шаг свертки фиксируется на уровне 1 пикселя, чтобы сохранить пространственное разрешение после свертки (шаг — это количество сдвигов пикселей по входной матрице).

Полносвязные слои: После сверточных блоков следуют три полносвязных слоя. Первые два полносвязных слоя имеют 4096 нейронов каждый, а последний слой состоит из 1000 нейронов, соответствующих количеству классов в наборе данных ImageNet. На выходе используется функция Softmax для предсказания вероятностей классов.

Нормализация и регуляризация: В модели применяется L2-регуляризация и дропаут (dropout) в полносвязных слоях для предотвращения переобучения.

Далее можно также выделить ряд преимуществ данной модели:

- Использование маленьких фильтров 3x3: Фильтры размером 3x3 позволяют сети захватывать более детализированные признаки изображения и обеспечивают большую глубину при меньшем количестве параметров, чем фильтры большего размера. Такая архитектура упрощает вычислительные операции, делая модель более эффективной.
- Глубокая архитектура: Большое количество слоев позволяет модели извлекать сложные и абстрактные признаки из изображений, что значительно повышает точность классификации.
- Преимущества модульной структуры: Разделение сети на блоки с последовательным увеличением количества фильтров и применением max-pooling слоев помогает постепенно уменьшать размерность данных, сохраняя при этом важные признаки.

VGG19 является одной из самых популярных моделей в области компьютерного зрения благодаря своей простой и эффективной архитектуре. Она широко используется в различных задачах, таких как классификация изображений, детекция объектов и сегментация, и служит основой для многих улучшенных архитектур глубокого обучения[12].

B. InceptionV3

InceptionV3 — это одна из самых известных и мощных моделей сверточных нейронных сетей, разработанных для задач классификации изображений. Она была представлена в 2015 году командой Google Research и является третьей версией серии Inception. Основной целью архитектуры Inception является повышение точности классификации при сохранении эффективности вычислений.

Далее представлена полная архитектура модели InceptionV3:

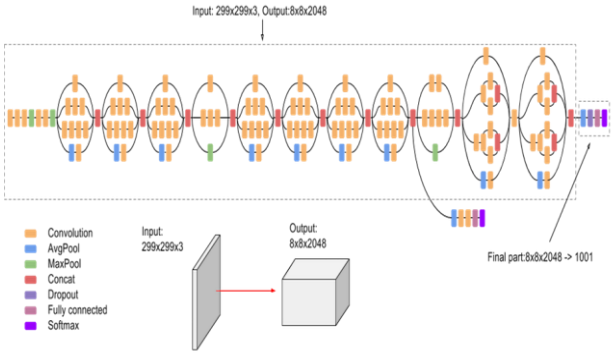


Рис.4 – архитектура сети InceptionV3

Архитектура InceptionV3 включает в себя несколько ключевых инноваций, которые улучшают производительность и эффективность модели.

Inception-модули:

- В основе архитектуры лежат так называемые Inception-модули, которые позволяют эффективно обрабатывать различные пространственные масштабы. Внутри каждого Inception-модуля используются сверточные фильтры разного размера (1x1, 3x3, 5x5), а также слои подвыборки (pooling).
- Это позволяет модели захватывать признаки на различных уровнях детализации и масштабов, что улучшает обобщающую способность сети.

Факторизация сверток:

Один из ключевых улучшений в InceptionV3 — факторизация сверток. Вместо использования больших сверточных фильтров (например, 5x5), модель разбивает их на комбинации более мелких фильтров (например, две последовательные свертки 3x3). Это позволяет значительно уменьшить количество параметров и вычислительные затраты, не теряя при этом качества извлечения признаков.

Асимметричные свертки:

- Далее несмотря на то, что более крупные свертки разлагаются на более мелкие свертки, вместо попыток факторизовать свертку, например, до свертки 2×2, была использована лучшая альтернатива, позволяющая сделать модель более эффективной, данной альтернативой стали асимметричные свертки. Асимметричные свертки имеют форму n×1. Итак, были заменены свертки 3×3 на свертку 1×3, за которой следовала свертка 3×1. Это то же самое, что сдвинуть двухслойную сеть с тем же рецептивным полем, что и в свертке 3×3.

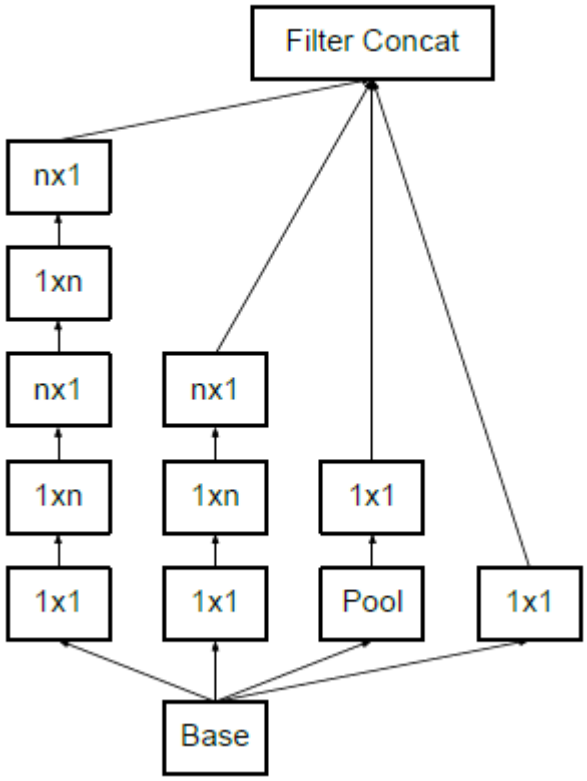


Рис.5 – структура асимметричной свертки

Ауксилиарные (вспомогательные) классификаторы:

- InceptionV3 также включает вспомогательные классификаторы, которые размещаются после промежуточных слоев сети. Эти классификаторы помогают ускорить процесс обучения и обеспечивают регуляризацию, что снижает риск переобучения.

Всего InceptionV3 состоит из 42 слоев. Модель начинается с нескольких стандартных сверточных слоев, которые уменьшают размерность входных данных и выделяют начальные признаки. Далее идет первый уровень Inception-модулей в количестве 3. Каждый такой модуль содержит свертки 1x1, 3x3 и 5x5, а также слои подвыборки (pooling). Затем следует второй уровень Inception-модулей в количестве 5. Данные модули включают в себя асимметричные свертки. После идет третий уровень Inception-модулей в количестве 2. Которые содержат еще более сложные комбинации факторизованных сверточных фильтров и слоев подвыборки. После Inception-модулей идут несколько стандартных слоев, включая глобальный слой подвыборки (global average pooling) и полностью связанный слой с 1000 нейронами[13].

Таким образом благодаря использованию факторизованных сверток и других оптимизаций, модель достигает высокой производительности при относительно низких вычислительных затратах.

IV. Оценка точности

Для сравнения эффективности моделей VGG19 и Inceptionv3 в данном исследовании были выбраны несколько наиболее распространенных для подобных задач метрик.

TP (True Positives) — изображения верно предсказанные как относящиеся к положительному классу, TN (True Negatives) — верно классифицированные изображения относящиеся к отрицательному классу, FP (False Positives) — количество ошибочно классифицированных изображений как положительные, иными словами относятся к отрицательным но модель допустила ошибку и определила их в положительный класс, FN (False Negatives) — изображения ошибочно классифицированные как отрицательные, то есть класс изображений положительный но определен моделью как отрицательный.

Precision (Точность): Precision показывает долю правильных положительных предсказаний среди всех предсказанных положительных случаев. Формула:

$$\textbf{Precision} = \frac{TP}{TP+FP} \quad (1)$$

Recall (Полнота): Recall измеряет долю правильно предсказанных положительных случаев среди всех реальных положительных случаев.

$$\textbf{Recall} = \frac{TP}{TP+FN} \quad (2)$$

F1 Score: Это гармоническое среднее Precision и Recall, которое дает сбалансированную оценку, особенно полезную при несбалансированных данных(когда одна из категорий ,например, больные листья представлена гораздо меньше, чем другая).

$$\textbf{F1} = 2 * \frac{Precision * Recall}{Precision+Recall} \quad (3)$$

Далее представлена таблица I с информацией о метриках моделей на тестовом датасете.

ТАБЛИЦА I. Результаты оценки моделей на тестовом датасете

	VGG19	**InceptionV3**
Precision	0.85	0.88
Recall	0.81	0.84
F1	0.83	0.86

Из данной таблицы видно что модель InceptionV3 справилась лучше чем VGG19 разница по метрике F1 составила 0.03. Тем не менее для данного датасета обе модели продемонстрировали достаточно хорошие в рамках данной задачи результаты, но пока можно сделать предварительный вывод о том что модель InceptionV3 лучше подходит для задачи распознавания больных растений.

Далее представлена таблица II с информацией о метриках моделей при использовании их на собственном наборе изображений, в количестве 20, из которых часть являлась новыми типами растений, что соответственно затрудняло задачу корректной классификации для обеих моделей.

ТАБЛИЦА II. Результаты оценки моделей на собственном датасете

	VGG19	**InceptionV3**
Precision	0.75	0.82
Recall	0.67	0.78
F1	0.71	0.80

Как видно из данной таблицы, разница на усложненных данных составила 0.09. Таким образом можно сделать вывод что модель InceptionV3 лучше справляется с задачей идентификации больных растений.

V. Заключение

Таким образом были рассмотрены две модели – VGG19 и InceptionV3, для определения больных растений лучше использовать InceptionV3, так как она превосходит по точности VGG19, что делает ее предпочтительной для дальнейшего применения и развития в агротехнических приложениях. Модели, подобные InceptionV3, могут быть интегрированы в системы мониторинга сельскохозяйственных культур, что позволит своевременно выявлять заболевания растений и принимать меры по их лечению. В качестве дальнейших исследований перспективным направлением является интеграция методов компьютерного зрения с другими технологиями, такими как интернет вещей (IoT) и дроны, для создания комплексных систем мониторинга сельскохозяйственных угодий.

Литература.

[1] Bazargani K, Deemyad T. Automation’s Impact on Agriculture: Opportunities, Challenges, and Economic Effects. Robotics. 2024; 13(2):33. https://doi.org/10.3390/robotics13020033

[2] Domingues T, Brandão T, Ferreira JC. Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey. Agriculture. 2022; 12(9):1350. https://doi.org/10.3390/agriculture12091350

[3] 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

[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] Sadekov R. N. et al. Road sign detection and recognition in panoramic images to generate navigational maps //2017 24th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS). – IEEE, 2017. – С. 1-5.

[6] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[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] 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.

[9] Solodov1, S.V., Mamai1, I.B. and Pronichkin2, S.V. (2022) IOPscience, IOP Conference Series: Earth and Environmental Science. Available at: https://iopscience.iop.org/article/10.1088/1755-1315/981/2/022007

[10] Martinelli, Federico & Scalenghe, Riccardo & Davino, Salvatore & Panno, Stefano & Scuderi, Giuseppe & Ruisi, Paolo & Villa, Paolo & Stroppiana, Daniela & Boschetti, Mirco & Goulart, Luiz & Davis, CristinaE & Dandekar, AbhayaM. (2014). Advanced methods of plant disease detection. A review. Agronomy for Sustainable Development. 35. 1-25. 10.1007/s13593-014-0246-1.

[11] Kour, Vippon Preet; Arora, Sakshi (2019), "PlantaeK: A leaf database of native plants of Jammu and Kashmir", Mendeley Data, V2, doi: 10.17632/t6j2h22jpx.2

[12] K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition", The 3rd International Conference on Learning Representations (ICLR2015), 2014, pp. 1-14.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe et. al. "Rethinking the Inception Architecture for Computer Vision", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2818-2826.

Классификация видов птиц при помощи компьютерного зрения

М. О. Левичкин
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1901074@edu.misis.ru

***Аннотация* — птицы являются неотъемлемой частью любой экосистемы и играют важную роль в природе. В свете этого применение передовых технологий для распознавания птиц становится ключевым инструментом в усилиях по мониторингу и защите их популяций. В данной работе рассматриваются различные решения с использованием нейронных сетей, а также проводится сравнение их возможностей в задаче классификации видов птиц при помощи компьютерного зрения, используя различные наборы данных. Целью данного исследования является выявление эффективных моделей, способных качественно классифицировать виды птиц. В ходе работы были выявлены сложности в поиске эффективных подходов. Исследование имеет значимость в контексте улучшения современных методов орнитологической диагностики и расширению знаний о биоразнообразии.**

Ключевые слова — Компьютерное зрение, Классификация изображений, Распознавание видов птиц, EfficientNet, VGG, CUB, Kaggle

I. Введение

Методы компьютерного зрения играют ключевую роль в развитии автоматизированных систем классификации видов птиц по изображениям. Они позволяют анализировать и обрабатывать изображения с высокой точностью и скоростью, что делает возможным эффективное распознавание различных характеристик птиц, таких как форма, размер, окраска и текстура оперативно и без необходимости вручную размечать данные [1]. Эти методы базируются на использовании различных алгоритмов компьютерного зрения, включая сверточные нейронные сети (CNN) [2], которые способны автоматически извлекать признаки из изображений и классифицировать их в соответствии с определенными критериями. Вместе с тем, разработка и применение таких систем имеет большое значение не только для научных исследований, но и для практических приложений, таких как мониторинг экосистем, охрана природы и биоразнообразия, а также контроль за воздействием человеческой деятельности на животный мир [3].

Птицы играют важную роль в поддержании баланса экосистем. Они не только помогают нам лучше понять мир природы, но и предоставляют ценную информацию о состоянии окружающей среды [4]. Орнитологи и учёные-экологи используют птиц для оценки состояния экосистем и выявления изменений в окружающей среде, включая уровень загрязнения. Поэтому точная классификация видов птиц имеет большое значение. Однако определение видов птиц по фотографиям представляет собой уникальную задачу, связанную с различиями в подвидах птиц, сложностью фона, условиями освещения и изменчивостью позы [5].

Учитывая вышеизложенные обстоятельства, в настоящей работе используются методы компьютерного зрения [6], [7], [8] для автоматизированной классификации видов птиц. В исследовании рассматриваются и сопоставляются результаты двух сверточных нейронных сетей в контексте определения видов птиц: VGG-16 [9] и EfficientNetB0 [10], применяемых в области глубокого обучения для классификации.

II. Наборы данных

В данной работе представлены 2 набора данных, взятых из открытых источников. Рассмотрим используемые наборы.

A. CUB-200-2011 [11]

Набор данных "CUB-200–2011" (Caltech-UCSD Birds-200–2011) состоит из более чем 11 000 изображений и 200 различных категорий, каждая из которых представляет отдельный вид птиц (рисунок 1).

Рис. 1. Примеры кадров различных видов птиц

Каждое изображение в датасете представлено в формате JPEG и сопровождается аннотацией, содержащей информацию о виде птицы, ее идентификаторе и различных атрибутах (рисунок 2). Набор данных обеспечивает разнообразие изображений, сделанных в различных условиях. Также птицы запечатлены в различных позах, что позволяет эффективно обучать модели.

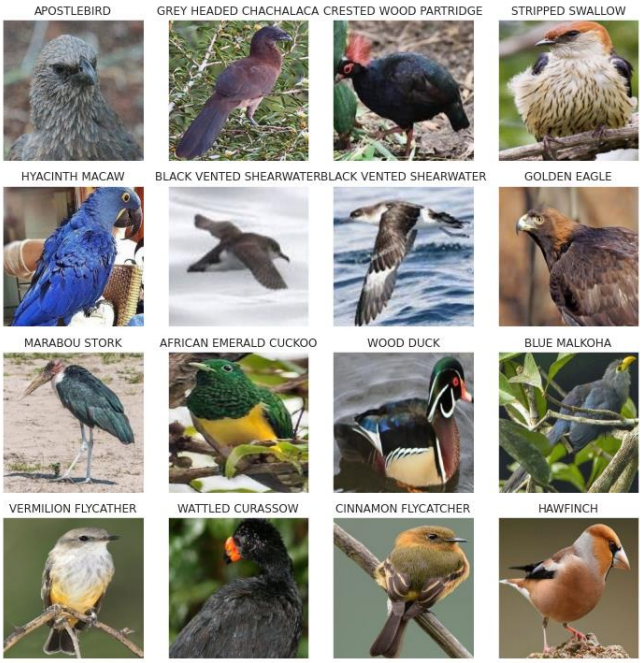


Рис. 2. Примеры классов птиц в наборе данных CUB-200–2011

Каждый вид птицы обладает своими уникальными внешними особенностями. Например, птицы-синицы и птицы-личинки могут иметь схожие размеры и окраску, что делает их сходство заметным на первый взгляд. Поскольку некоторые виды птиц крайне схожи с другими это, в значительной мере, затрудняет классификацию. На примере 12 видов птиц видно, как разные классы птиц могут быть похожи внешне, но на самом деле быть в разных классах и наоборот – быть внешне различны, но при этом относиться к одному классу (рисунок 3).

Рис. 3. Примеры схожих птиц: (а), (б), (в) - внешнее сходство при различии видов, (г), (д), (е) – внешнее различие одних и тех же видов [1]

B. Дополненный набор данных

Данный набор данных является дополненным, по отношению к набору данных «CUB-200–2011». Дополненный 20 видами птиц из открытого набора данных на платформе Kaggle. Набор данных «CUB-200–2011» сам по себе является обширным набором. Дополнительные изображения необходимы для повышения качества проверки нейронных сетей. Примеры дополнительных изображений отражены на рисунке 4.

Рис. 4. Некоторые образцы из дополненного набора

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

В данной работе для классификации видов птиц применялись две нейронных сети: VGG-16 и EfficientNetB0. Эти модели обладают способностью эффективно выделять высокоуровневые признаки из изображений [12], что является важным для достижения высокой точности при решении сложных задач классификации объектов, в частности птиц, обладающих разнообразием визуальных характеристик

A. VGG-16

Для классификации видов птиц была использована сверточная нейросетевая модель VGG-16. Модель была разработана в 2014 году исследователями из Оксфордского университета и получила известность благодаря своим высоким результатам в конкурсе ILSVRC

(ImageNet Large Scale Visual Recognition Challenge) в том же году [13].

Основные особенности VGG-16:

- Архитектура: сеть состоит из 16 слоев, включая 13 сверточных слоев и 3 полносвязных слоя. Она использует очень маленькие сверточные фильтры размером 3x3. Архитектура сети относительно проста и однородна, поскольку используются только сверточные слои и полносвязные слои.
- Глубина: VGG-16 считается одной из первых действительно глубоких сверточных нейронных сетей с большим количеством слоев (16 для VGG-16).
- Сверточные фильтры: вместо использования больших фильтров, VGG-16 использует последовательность нескольких маленьких фильтров 3x3, что повышает нелинейность сети.

Для классификации используется версия архитектуры VGG-16, которая доступна в TensorFlow. Модель дообучается на изображениях из набора данных CUB-200–2011 и на дополнительном открытом наборе данных Kaggle. Для улучшения обучения модели осуществлялась аугментация изображений с помощью модуля ImageDataGenerator из библиотеки Keras. В качестве оптимизатора используется Adam с параметрами по умолчанию. В качестве функции потерь используется кросс-энтропия, которая в сочетании с softmax активацией позволяет эффективно классифицировать виды птиц на изображениях.

Архитектура модели представлена на рисунке 5.

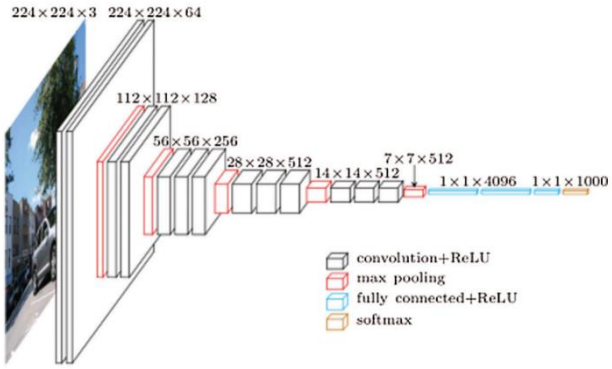


Рис. 5. Архитектура модели VGG-16 [14]

B. *EfficientNetB0*

EfficientNetB0 — это архитектура CNN, разработанная для достижения высокой точности и эффективности в задачах классификации изображений. Исследователи в работе [10] предложили новый метод комбинированного масштабирования для увеличения масштаба CNN принципиальным образом. Сверточные нейронные сети могут масштабироваться путем изменения глубины сети (количества слоев), ширины сети (количества каналов) или увеличения разрешения входного изображения, что позволяет сети извлекать более детализированные характеристики изображения. Архитектура сети использует немного более крупные сверточные блоки (MBConv), которые можно увидеть на рисунке 6.

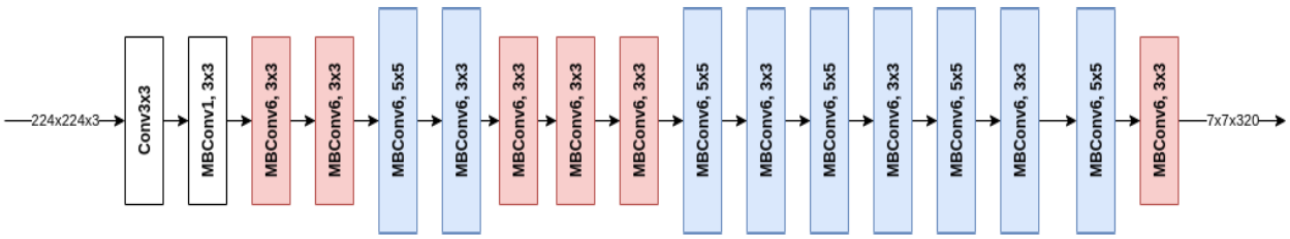


Рис. 6. Архитектура EfficientNetB0 CNN. В EfficientNetB0 используется несколько более крупная мобильная свертка (MBConv)

Для классификации используется версия архитектуры EfficientNetB0, которая доступна в TensorFlow. Модель дообучается на изображениях из набора данных CUB-200–2011 и на дополнительном открытом наборе данных Kaggle. Для улучшения обучения модели тажке осуществлялась аугментация изображений. В качестве оптимизатора используется Adam с параметрами по умолчанию. В качестве функции потерь используется кросс-энтропия в сочетании с softmax активацией.

IV. Сравнение

Сравним два описанных подхода. Для оценки качества работы подходов важно рассмотреть такие показатели как, например, F1-мера, чтобы получить более полное представление о производительности моделей. Введём следующие величины:

- TP – модель правильно классифицирует образец как принадлежащий к классу, когда на самом деле он принадлежит к этому классу.
- FP – модель неправильно классифицирует образец как принадлежащий к классу, когда на самом деле он не принадлежит к этому классу
- FN – модель неправильно классифицирует образец как не принадлежащий к классу, когда на самом деле он принадлежит к этому классу.

Стоит отметить, что TN в задачах классификации, особенно в многоклассовых, не всегда рассматривают, так как они относятся к отрицательным прогнозам, что не всегда применимо к каждому классу. По введённым величинам строятся такие функции оценок, как:

- $Precision = \frac{TP}{TP+FP}$ – доля истинно положительных результатов (TP, правильных предсказаний) от общего количества релевантных результатов, т. е. сумма TP и ложноположительных результатов (FP). Для задач многоклассовой классификации Precision усредняется по классам;
- $Recall = \frac{TP}{TP+FN}$ – доля TP от общего количества TP и ложноотрицательных результатов (FN). Для задач многоклассовой классификации Recall усредняется по всем классам;
- $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$ – среднее гармоническое значение точности и отзыва. Для задач многоклассовой классификации F1 усредняется по всем классам.

F1_score для каждой модели отображена в таблице 1:

ТАБЛИЦА I. Оценка F1_Score для каждой модели

Модель	**F1_Score**
EfficientNetB0	0.87
VGG-16	0.72

Лучшие результаты имеет модель EfficientNetB0. Точность модели составляет 0.87, что означает, что 87% положительных прогнозов, сделанных моделью, были правильными. Модель VGG-16 показала результаты несколько хуже по сравнению с моделью EfficientNetB0.

Стоит также оценить точность EfficientNetB0 для каждого вида птиц. В связи с тем, что классов крайне много, было принято решение, для улучшения визуализации, отобразить точность в виде таблицы для первых 50 видов (таблица 2).

ТАБЛИЦА II. Точность EfficientNetB0 для 50 видов

№	**Вид птицы**	**Точность %**	**№**	**Вид птицы**	**Точность %**
1	BLACK FOOTED ALBATROSS	87	26	YELLOW BREASTED CHAT	57
2	LAYSAN ALBATROSS	71	27	EASTERN TOWHEE	77
3	SOOTY ALBATROSS	82	28	CHUCK WILL WIDOW	100
4	GROOVE BILLED ANI	93	29	BRANDT CORMORANT	90
5	CRESTED AUKLET	96	30	RED FACED CORMORANT	65
6	PARAKEET AUKLET	87	31	PELAGIC CORMORANT	83
7	RHINOCEROS AUKLET	100	32	BRONZED COWBIRD	85
8	BREWER BLACKBIRD	70	33	ANDEAN SISKIN	85
9	RED WINGED BLACKBIRD	97	34	ANHINGA	90
10	RUSTY BLACKBIRD	49	35	ANIANIAU	66
11	YELLOW HEADED BLACKBIRD	87	36	ANNAS HUMMINGBIRD	84
12	BOBOLINK	92	37	ANTBIRD	90
13	INDIGO BUNTING	89	38	ANTILLEAN EUPHONIA	78
14	LAZULI BUNTING	84	39	APAPANE	89
15	PAINTED BUNTING	100	40	APOSTLE BIRD	67
16	CARDINAL	100	41	ARARIPE MANAKIN	90
17	SPOTTED CATBIRD	94	42	ASHY STORM PETREL	78
18	GRAY CATBIRD	87	43	ASHY THRUSH BIRD	73
19	AMERICAN FLAMINGO	97	44	ASIAN CRESTED IBIS	96
20	AMERICAN GOLDFINCH	72	45	ASIAN DOLLARD BIRD	77
21	AMERICAN KESTREL	94	46	ASIAN GREEN BEE EATER	87
22	AVADAVAT	57	47	ASIAN OPENBILL STORK	84
23	AZARAS SPINETAIL	76	48	AUCKLAND SHAQ	83

24	AZURE BREASTED PITTA	100	49	AUSTRAL CANASTERO	88
25	AZURE JAY	92	50	AUSTRALASIAN FIGBIRD	89

Некоторые виды птиц, такие как риноцеросный чистик и расписная овсянка, были успешно классифицированы моделью EfficientNetB0 с идеальной точностью. Это говорит о том, что эти виды имеют уникальные и хорошо заметные характеристики, которые модель легко распознаёт и использует для классификации. Виды, классифицированные с точностью ниже 70%, требуют особого внимания, поскольку низкая точность может указывать на наличие похожих между собой видов, что создаёт путаницу для модели. Тем не менее другие классы показывают точность, близкую к 100%, что является хорошим результатом. В целом модель демонстрирует высокую точность.

Исходя из этих результатов, можно сделать вывод, что модель EfficientNetB0 предпочтительнее модели VGG-16 в контексте классификации изображений птиц. Ее более сбалансированные и стабильные результаты оценки производительности делают ее более надежной и эффективной для данной задачи.

V. Заключение

В данной работе были исследованы различные методы классификации видов птиц с использованием компьютерного зрения и сверточных нейронных сетей. Были рассмотрены два набора данных: CUB-200–2011 и дополненный набор данных, включающий дополнительные 20 видов птиц из открытого источника.

Сравнение результатов показало, что модель EfficientNetB0 превзошла VGG-16 по показателю F1-меры, достигнув значения 0.87 против 0.72 у VGG-16. Исходя из полученных результатов, можно сделать вывод, что модель EfficientNetB0 является предпочтительной для задачи классификации видов птиц по сравнению с VGG-16, демонстрируя более сбалансированные и стабильные показатели производительности.

В целом, результаты исследования показывают, насколько важно использовать современные нейронные сети для распознавания в таких областях, как мониторинг экосистем и охрана природы. Также исследование подчёркивает важность выбора оптимальной архитектуры нейронной сети для каждой конкретной задачи.

Литература

[1] Atanbori, J., Duan, W., Murray, J. Automatic classification of flying bird species using computer vision techniques. Pattern Recognition Letters, 81, pp.53-62.

[2] Толстенко Л.С., Клейменов А.А., Али Б., Крынецкая Г.С., Коробков А.А. Анализ нейронных сетей для детектирования светофоров на изображениях // известия института инженерной физики. — 2023. — № 2(68). — С. 59-65.

[3] Weinstein, B.G., 2018. A computer vision for animal ecology. Journal of Animal Ecology, 87(3), pp.533-545.

[4] Sekercioglu, C.H., 2006. Increasing awareness of avian ecological function. Trends in ecology & evolution, 21(8), pp.464-471.

[5] Branson, S., Van Horn, G., Belongie, S. Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952.

[6] Sitepu, A.C., Liu, C.M., Sigiro, 2022. A convolutional neural network bird's classification using north american bird images. Journal of Health Sciences, 6(S2), pp.15-067.

[7] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95.10.21146/0042-8744-2022-3-93-105.

[8] I. Shazzadul, K. Sabit, A. Habibullah. "Bird Species Classification from an Image Using VGG-16 Network", the 2019 7th International Conference, 2019, pp. 38-42

[9] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[10] Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[11] CUB-200-11 birds images data set, avaliable at: https://www.kaggle.com/datasets/wenewone/cub2002011/data (Accessed: October 01, 2023).

[12] Berg, T., Liu, J., Lee, S.W. Birdsnap: Large-scale fine-grained visual categorization of birds, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE. pp. 2019–2026

[13] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14] Sriram, G., Babu, T.R., Praveena, R. and Anand, J.V., 2022. Classification of Leukemia and Leukemoid Using VGG-16 Convolutional Neural Network Architecture. Molecular & Cellular Biomechanics, 19(2).

[15] Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[16] H. Amin, A. Darwish, A. E. Hassanien and M. Soliman, "End-to-End Deep Learning Model for Corn Leaf Disease Classification," in IEEE Access, vol. 10, pp. 31103-31115, 2022

[17] International Joint Conference on Neural Networks (IJCNN), pp. 1–8, July 2018.

[18] W. Man, Y. Ji, and Z. Zhang, "Image classification based on improved random forest algorithm," in 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 2018, pp. 346–350.

[19] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618-626

[20] G. Amato and F. Falchi, "kNN based image classification relying on local feature similarity," in Proceedings of the Third International Conference on SImilarity Search and APplications - SISAP '10, 2010, p. 101.

[21] S. Dave, "Image Classification Algorithm based on MultiFeature Extraction and KNN Classifier," Int. J. Adv. Eng. Res. Dev., vol. 4, no. 6, 2017.

[22] 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.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in Advances in neural information processing systems, pp. 91–99, 2015.

[24] R. Hamzah, H. Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms", Journal of Sensors, 2016, pp. 1-23.

[25] J. Redmon, A. Farhadi. "YOLOv3: An Incremental Improvement", ArXiv abs/1804.02767, 2018, pp. 1-6.

[26] 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.

[27] W. Liu, D. Anguelov, D. Erhan et al. "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision, 2015, pp. 1-17.

[28] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, The 3rd International Conference on Learning Representations (ICLR2015), 2014, pp. 1-14.

[29] A. G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Application.,” Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861, 2017, pp. 1-9.

[30] T. Lin, M. Maire, S. J. Belongie, Hays et. al. “Microsoft COCO: Common Objects in Context. European Conference on Computer Vision”, Computer Vision (ECCV2014), 2014, vol. 8693, pp. 740-755.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe et. al. “Rethinking the Inception Architecture for Computer Vision”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2818-2826.

[32] 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.

[33] 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.

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

[35] 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.

28.23.15
1707
01.02.EP

Human pose estimation на изображениях асан в йоге

Леонов Иван Юрьевич
Университет МИСИС
m2311081@edu.misis.ru

***Аннотация* — одной из ключевых задач компьютерного зрения в настоящий момент является Pose estimation, или оценка позы. Этот процесс заключается в определении координат ключевых точек тела. Технологии по оценке позы нашли широкое применение в различных областях, в частности в спорте. Особенно такие технологии полезны в физических практиках на подобии йоги, так как правильное положение тела при выполнении асан (поз) имеет ключевое значения для достижения желаемого результата. В данной статье рассматриваются два инструмента для оценки поз MMPose и YOLOv8-pose и их применение на двух датасетах с изображениями асан.**

***Ключевые слова* — оценка позы, асан, MMPose, YOLOv8-pose, ключевые точки тела, PCK, OKS**

I Введение

Pose estimation, или оценка позы, является одной из ключевых задач компьютерного зрения, направленной на определение положения человеческого тела в изображении или видео [1]. Этот процесс включает в себя распознавание и определение координат ключевых точек тела, таких как суставы и конечности, что позволяет понять, как именно расположено тело в пространстве [2, 3]. В последние годы, благодаря развитию глубокого обучения и улучшению алгоритмов, технология оценки позы достигла значительных успехов и нашла широкое применение в различных областях, включая медицину, спорт, анимацию и виртуальную реальность [4, 5, 6, 7, 8].

Йога, как древняя практика, направлена на гармонизацию тела и разума через выполнение определенных поз, или асан. Правильное выполнение асан имеет ключевое значение для достижения желаемого эффекта, будь то улучшение гибкости, укрепление мышц или расслабление. Однако контроль за правильностью выполнения асан может быть затруднен, особенно для начинающих, у которых нет возможности посещать занятия под руководством опытного инструктора.

В этом контексте технологии оценки позы открывают новые возможности для автоматического мониторинга и анализа выполнения асан. Используя камеры и специализированные алгоритмы, можно в реальном времени отслеживать положение тела, предоставлять обратную связь о правильности выполнения того или иного асана [9, 10].

В рамках статьи будет проведен обзор двух инструментов для оценки поз, а именно MMPose и YOLOv8-pose, и рассмотрено их применение на изображениях асан в йоге.

II MMPose

MMPose – это открытая библиотека для оценки позы, разработанная в рамках проекта OpenMMLab, которая предоставляет мощные и гибкие инструменты для решения задач оценки позы человека и животных. иерархические признаки из входных изображений [11].

Одной из ключевых особенностей MMPose является его модульная и гибкая архитектура. Фреймворк поддерживает множество моделей и алгоритмов оценки позы, включая как двухмерные (2D), так и трехмерные (3D) методы. Это позволяет пользователям выбирать оптимальные решения для различных задач и сценариев применения.

В основе MMPose лежат различные архитектуры нейронных сетей, используемые в качестве backbone для извлечения признаков из изображений. Эти архитектуры включают ResNet, HRNet и другие популярные модели, которые обеспечивают высокую точность и производительность.

Компонент neck объединяет признаки, извлеченные из различных слоев базовой сети, чтобы создать более информативное представление данных. Включает в себя различные операции, такие как объединение, свертка, пулинг и активация, которые помогают улучшить качество признаков.

На этапе head происходит непосредственно предсказание позы на основе извлеченных признаков. MMPose поддерживает различные типы heads, включая Heatmap-based heads и Regression-based heads. Heatmap-based подходы создают тепловые карты для каждого ключевого сустава, которые затем используются для определения точных координат суставов. Regression-based подходы напрямую предсказывают координаты суставов, что может быть более эффективно в некоторых случаях.

Декодеры преобразуют выходные данные головных сетей в окончательные предсказания позы в формате координат ключевых точек. Это может включать в себя дополнительные шаги обработки, такие как интерполяция, фильтрация шума и нормализация координат

Для обучения моделей используется множество различных функций потерь, которые помогают улучшить точность предсказаний. Одной из распространенных функций потерь является Mean Squared Error (MSE), используемая для обучения тепловых карт. Для регрессионных моделей могут применяться такие функции, как Smooth L1 Loss, которая сочетает в себе преимущества L1 и L2 потерь.

Для повышения обобщающей способности моделей используется широкий спектр методов аугментации данных. MMPose поддерживает такие техники, как случайное изменение масштаба, вращение, изменение яркости и контраста, что помогает моделям быть более устойчивыми к разнообразным условиям съемки.

MMPose поддерживает множество форматов данных и аннотаций, что упрощает интеграцию с различными датасетами. Это включает в себя популярные наборы данных, такие как COCO, MPII и другие, что позволяет быстро начать обучение моделей на различных типах данных.

MMPose оптимизирован для высокой производительности, благодаря использованию эффективных алгоритмов и оптимизированной реализации на GPU.

III YOLOv8-POSE

YOLOv8-pose – это модель компьютерного зрения, разработанная для задач обнаружения объектов и их частей тела. Эта модель является развитием серии YOLO от Ultralytics [12, 13].

В качестве backbone в YOLOv8-pose используется модифицированный вариант ResNet-50, который был оптимизирован для работы с большим количеством параметров и слоев.

Компонент neck в YOLOv8-pose представлен в виде FPN (Feature Pyramid Network), который создает пирамиду признаков с различными масштабами, что позволяет модели обрабатывать объекты разного размера [14].

На этапе head используется модуль DLA (Dynamic Layer Aggregation), который объединяет информацию от различных слоев neck и использует различные стратегии для предсказания объектов и их частей тела.

Декодеры в YOLOv8-pose выполняют важную роль в преобразовании предсказаний модели в формат, удобный для дальнейшего использования. Каждый декодер специализируется на определенном типе объекта или части тела. Например, есть отдельные декодеры для головы, рук, ног и т.д. Они преобразуют предсказания модели в координаты и классы объектов.

YOLOv8-Pose оснащен мощными инструментами для обучения и оценки моделей. Это включает использование различных функций потерь, таких как L2 loss для регрессии координат и cross-entropy loss для тепловых карт, а также метрики, такие как mean Average Precision (mAP) для оценки точности модели.

Модель поддерживает множество стандартных датасетов для оценки позы, таких как COCO, MPII и другие, что облегчает обучение и тестирование на разнообразных данных.

IV ПОДГОТОВКА ДАТАСЕТА

Для оценки качества оценки поз MMPose и YOLOv8-pose были подготовлены два датасета из 30 изображений. В первом случае был изображён асан Вирабхадрасана 2 (поза Героя), а во втором – Уткатасана (поза Богини) [15].

Рис. 1 – Асан Вирабхадрасана 2

Рис. 2 – Асан Уткатасана

С помощью Label Studio все изображения были размечены [16]. В разметке использовались 17 ключевых точек тела:

1. нос;
2. левый глаз;
3. правый глаз;
4. левое ухо;
5. правое ухо;
6. левое плечо;
7. правое плечо;
8. левый локоть;
9. правый локоть;
10. левое запястье;
11. правое запястье;
12. левое бедро;
13. правое бедро;
14. левое колено;
15. правое колено;

16. левая лодыжка;

17. правая лодыжка.

V Сравнение MMPose и YOLOv8-pose в задаче Pose estimation на изображениях асан

Перед оценкой моделей на двух датасетах был произведен инференс на изображении асана Васиштхасана.

Рис. 3 – Ключевые точки тела, полученные с использованием MMPose

Рис. 4 – Ключевые точки тела, полученные с использованием YOLOv8-pose

При визуальной оценке можно отметить, что оба инструмента справляются достаточно хорошо с поставленной задачей при минимальном количестве неточностей.

Для того, чтобы количественно оценить работу моделей на двух датасетах были использованы метрики PCK и OKS [17, 18, 19].

PCK (Percentage of Correct Keypoints) – метрика вычисляет процент правильных ключевых точек, где ключевая точка считается правильной, если она находится на расстоянии меньше некоторого порога от истинного положения.

Так как при визуальной оценке было выявлено, что обе модели достаточно хорошо определяют ключевые точки тела, то было принято решение установить пороговое значения в районе 0,05 для повышения точности метрики.

OKS (Object Keypoint Similarity) – метрика вычисляет сходство между предсказанными и истинными ключевыми точками, принимая во внимание масштабы и неопределенность каждой ключевой точки.

$$OKS = \sum_i \exp\left(\frac{-d_i^2}{2s^2k_i^2}\right), \quad (1)$$

где

- d_i – евклидово расстояние между эталонными данными и прогнозируемой ключевой точкой;
- s – квадратный корень из площади сегмента;
- k_i – константа ключевой точки, контролирующая снижение.

Чем ближе метрики к 100%, тем лучше качество модели.

ТАБЛИЦА I – Метрики PCK и OKS

Датасет	Метрика	MMPose	YOLOv8-pose
Первый	**PCK**	97,25%	97,65%
	OKS	87,91%	88,25%
Второй	**PCK**	95,69%	89,61%
	OKS	88, 08%	88,37%

Из таблицы 1 видно, что обе модели показали примерно равные хорошие показатели по обеим метрикам. Это может свидетельствовать, что и MMPose, и YOLOv8-pose обладают высоким качеством определения ключевых точек тела.

YOLOv8-pose оказалась чуть лучше практически во всех случаях, кроме метрики PCK на втором датасете. Таким образом, YOLOv8-pose в рамках задачи human pose estimation на изображениях асан в йоге является предпочтительной моделью.

VI Выводы

В данной статье были рассмотрены инструменты для оценки поз MMPose от OpenMMLab и YOLOv8-pose от Ultralytics.

Данные модели сравнивались на двух датасетах асан Вирабхадрасана 2 и Уткатасана с помощью метрик PCK и OKS.

Обе модели показали высокие результаты. Однако если выбирать между ними, то предпочтение стоит отдать YOLOv8-pose. Продукт от Ultralytics показал себя чуть лучше для задачи human pose estimation на изображениях асан в йоге и при всём этом более прост в использовании.

Список литературы

[1] Zabihifar, S.H., Semochkin, A., Seliverstova, E., & Efimov, A.R. (2021). Unreal mask: one-shot multi-object class-based pose estimation

for robotic manipulation using keypoints with a synthetic dataset. Neural Computing and Applications, 33, 12283 - 12300.

[2] 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).

[3] N., S., Guzhva., B., Ali., R., N., Sadekov., A., V., Sholokhov. (2023). Evaluating the Accuracy of Tram Positioning System in High-Rise Building Environment Using Data from Visual Geoinformation Systems. 1-5.

[4] Руководство по Human Pose Estimation [Электронный ресурс]. – Ресурс доступа: https://habr.com/ru/articles/687728/ (дата обращения: 20.05.2024).

[5] Human Pose Estimation with Deep Learning – Ultimate Overview in 2024 [Электронный ресурс]. – Ресурс доступа: https://viso.ai/deep-learning/pose-estimation-ultimate-overview/ (дата обращения: 20.05.2024).

[6] Pose Estimation Algorithms: History and Evolution [Электронный ресурс]. – Ресурс доступа: https://blog.roboflow.com/pose-estimation-algorithms-history/ (дата обращения: 20.05.2024).

[7] Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5686-5696.

[8] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., & Zhang, L. (2019). HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5385-5394.

[9] Йога глазами дата-сайентиста: как мы строили computer vision в мобильном приложении [Электронный ресурс]. – Ресурс доступа: https://habr.com/ru/articles/555162/ (дата обращения: 20.05.2024).

[10] Using Human Pose Estimation in Fitness & Rehab Therapy Apps [Электронный ресурс]. – Ресурс доступа: https://mobidev.biz/blog/human-pose-estimation-technology-guide (дата обращения: 20.05.2024).

[11] Welcome to MMPose’s documentation! [Электронный ресурс]. – Ресурс доступа: https://mmpose.readthedocs.io/en/latest/ (дата обращения: 22.05.2024).

[12] Ultralytics YOLO Docs [Электронный ресурс]. – Ресурс доступа: https://docs.ultralytics.com/ (дата обращения: 22.05.2024).

[13] YOLO: Algorithm for Object Detection Explained [+Examples] [Электронный ресурс]. – Ресурс доступа: https://www.v7labs.com/blog/yolo-object-detection (дата обращения: 22.05.2024).

[14] Illarionova S, Shadrin D, Tregubova P, Ignatiev V, Efimov A, Oseledets I, Burnaev E. A Survey of Computer Vision Techniques for Forest Characterization and Carbon Monitoring Tasks. Remote Sensing. 2022; 14(22):5861.

[15] Yoga Pose Detection [Электронный ресурс]. – Ресурс доступа: https://www.kaggle.com/code/aayushmishra1512/yoga-pose-detection/input (дата обращения: 23.05.2024).

[16] Open Source Data Labeling | Label Studio [Электронный ресурс]. – Ресурс доступа: https://labelstud.io/ (дата обращения: 23.05.2024).

[17] Xiao, B., Wu, H., & Wei, Y. (2018). Simple Baselines for Human Pose Estimation and Tracking. ArXiv, abs/1804.06208.

[18] A Comprehensive Guide to Human Pose Estimation [Электронный ресурс]. – Ресурс доступа: https://www.v7labs.com/blog/human-pose-estimation-guide (дата обращения: 23.05.2024).

[19] Keypoint Evaluation [Электронный ресурс]. – Ресурс доступа: https://cocodataset.org/#keypoints-eval (дата обращения: 23.05.2024).

Эффективность различных архитектур нейронных сетей в задаче распознавания медицинских масок

А. Г. Лойко
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2314262@edu.misis.ru

Я. О. Канунникова
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1805450@edu.misis.ru

Аннотация — В последние годы использование медицинских масок стало важной мерой для предотвращения распространения инфекционных заболеваний. Эффективное распознавание наличия масок на лицах в реальном времени играет ключевую роль в обеспечении безопасности общественных мест. В данной статье проводится анализ современных методов детекции медицинских масок, основанных на нейронных сетях. Рассматриваются различные подходы машинного обучения, включая использование таких инструментов и библиотек, как OpenCV, Keras, TensorFlow.. Исследование демонстрирует, как интеграция компьютерного зрения и нейронных сетей способствует повышению точности и скорости распознавания масок. Результаты анализа показывают, что применение передовых технологий позволяет значительно улучшить реализацию систем безопасности, основанных на детекции масок.

Ключевые слова — detection, machine learning, neural networks, OpenCV, Keras, TensorFlow, computer vision, YOLOv7

I. Введение

Последние годы пандемии COVID-19 сделали необходимым широкое использование медицинских масок для снижения распространения вируса. Это привлекло внимание к разработке автоматизированных систем, способных распознавать ношение масок на лицах в реальном времени. Появление и развитие технологий машинного обучения и глубоких нейронных сетей (ГНС) предоставило мощные инструменты для решения этой задачи, позволяя создавать высокоточные и эффективные системы детекции масок.

Настоящая статья посвящена анализу различных методов распознавания медицинских масок с использованием продвинутых технологий глубокого обучения, таких как библиотеки Keras и TensorFlow. Эти инструменты предоставляют мощные средства для построения и обучения моделей нейронных сетей, обеспечивая высокую скорость разработки и гибкость настройки алгоритмов. Особое внимание уделяется также использованию новой версии алгоритма YOLO (You Only Look Once) – YOLOv7, известного своей высокой производительностью в задачах детекции объектов в реальном времени.

В ходе исследования будут рассмотрены и сравнены различные подходы к распознаванию медицинских масок. Сравнительный анализ включает оценку качества распознавания, точности, скорости работы моделей и их применимости в реальных условиях. Это позволяет выявить сильные и слабые стороны каждого метода и выбрать наиболее эффективное решение для практического использования.

Таким образом, цель данной статьи – предоставить детальный обзор существующих методов распознавания медицинских масок с акцентом на использовании библиотек Keras и TensorFlow, а также YOLOv7, и провести сравнительный анализ их эффективности в различных аспектах.

II. Наборы данных и модели

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. Face-Mask-Detection

Датасет Face-Mask-Detection представляет собой тщательно подобранную коллекцию изображений, предназначенную для обучения и тестирования моделей детекции наличия медицинских масок на лицах. Датасет включает в себя следующие ключевые особенности:

Всего в наборе данных содержится 4095 изображений, из них 2165 фотографий людей, носящих маски и 1930 фотографий людей без масок.

Все изображения сохранены в формате .jpg, что обеспечивает их совместимость с большинством инструментов и библиотек для обработки изображений. Пример изображений людей в масках представлен на рисунке 1.

Все фотографии имеют очищенные фоны, что минимизирует влияние внешних факторов и фоновую шумовую информацию. Это позволяет моделям фокусироваться исключительно на распознавании наличия маски на лице, что особенно важно для повышения точности детекции.

Рисунок 1 – Пример датасета с людьми в масках

Датасет охватывает разнообразные условия съёмки, разнообразие ракурсов и освещения, а также включает в себя различные выражения лиц, что делает его

пригодным для обучения моделей, способных эффективно работать в реальных условиях.

Данный датасет является качественным ресурсом для исследований и разработок в области компьютерного зрения и машинного обучения, обеспечивая необходимую основу тестирования высокоточных систем детекции масок.

B. Medical Masks Dataset Images TFRecords

Датасет Medical Masks Dataset Images TFRecords, доступный на платформе Kaggle, представляет собой богатую коллекцию высококачественных изображений, предназначенных для обучения и оценки моделей детекции и классификации людей в медицинских масках. Основные характеристики датасета включают:

Датасет состоит из 1139 фотографий людей и групп людей в медицинских масках. Все изображения выполнены в формате .jpg и характеризуются высоким качеством. Это обеспечивает детальное представление лиц и масок, что делает датасет пригодным для задач компьютерного зрения и глубокого обучения.

Изображения упакованы в формате TFRecords, который оптимизирован для эффективного хранения и чтения данных в больших масштабах. Этот формат широко используется в TensorFlow и других фреймворках глубокого обучения для ускорения процессов чтения и обработки данных [1].

Датасет размечен, что означает, что каждая фотография сопровождается аннотациями, указывающими на присутствие или отсутствие маски на лице(ах). Разметка может включать координаты ограничивающих рамок (bounding boxes) вокруг лиц с масками, что позволяет моделям не только классифицировать, но и локализовать объекты на изображениях [2]. Пример изображений приведен на рисунке 2.

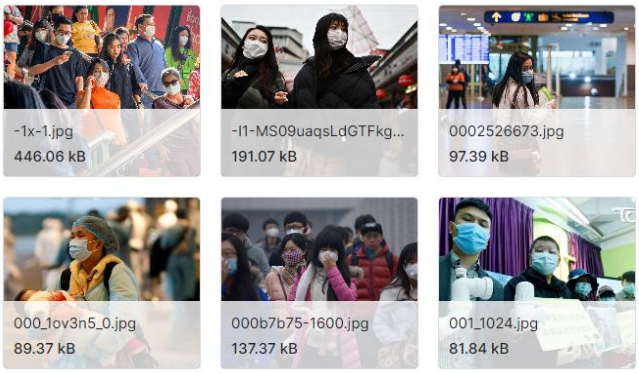


Рисунок 2 – Пример датасета людей в масках

C. Handmade dataset

Для конечно проверки был использован собственный собранный датасет , пример из него можно увидеть на рисунке 3.

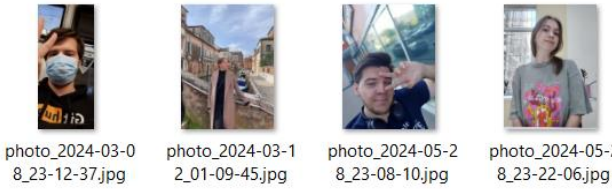


Рисунок 3 – Распределение датасета Emotional Detection

D. MobileNetV2.

Архитектура первой модели для обнаружения медицинских масок на лицах основана на использовании сверточной нейронной сети MobileNetV2 в сочетании с настраиваемым классификатором [3]. Данный подход обеспечивает высокую точность и эффективность распознавания, при этом оставаясь вычислительно экономичным. Основные компоненты архитектуры и их функции описаны ниже.

MobileNetV2 представляет собой легковесную и высокоэффективную архитектуру свёрточной нейронной сети, разработанную для мобильных и встроенных приложений. В этой модели используется версия MobileNetV2, предварительно обученная на наборе данных ImageNet. При этом верхний полносвязный слой (head FC layer) был удален, что позволяет использовать сеть в качестве эффективного экстрактора признаков.

На основе выходных данных MobileNetV2 был построен настраиваемый классификатор, включающий следующие слои:

1. Средний пуллинг (AveragePooling2D) выполняет операцию среднего пуллинга с размером ядра 7x7, что помогает уменьшить размерность выходных данных и выделить наиболее важные признаки.

2. Слой сглаживания (Flatten) позволяет преобразовать многомерный тензор признаков в одномерный массив, что необходимо для последующих полносвязных слоев.

3. Полносвязный слой (Dense) с 128 нейронами и функцией активации ReLU добавлен для выполнения нелинейного преобразования признаков, способствует выявлению сложных зависимостей в данных.

4. Слой Dropout (Dropout) с коэффициентом 0.5 добавлен для предотвращения переобучения модели, путем случайного отключения половины нейронов на каждом шаге обучения.

5. Выходной полносвязный слой (Dense содержит 2 нейрона с функцией активации Softmax, что позволяет модели выполнять бинарную классификацию – обнаружение лиц с маской и без маски.

Схему архитектуры можно увидеть на рисунке 4.

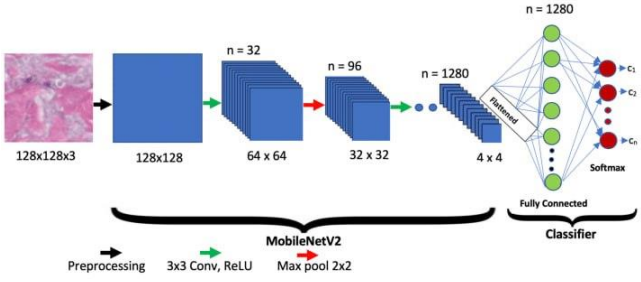


Рисунок 4 – Архитектура нейросети MobileNetV2

Процесс обучения модели включал следующие этапы:

1. Подготовка данных. Использовался ImageDataGenerator для расширения данных за счет различных аугментаций, таких как вращение, масштабирование, сдвиги по ширине и высоте, сдвиги по сдвигу, горизонтальные отражения и заполнение;

2. Компиляция модели. Модель была скомпилирована с использованием оптимизатора Adam и функции потерь binary_crossentropy. Начальная скорость обучения была установлена на 1e-4;

3. Обучение проводилось на протяжении 20 эпох с использованием батчей размера 32. Модель была обучена на 80% данных, оставляя 20% для валидации;

4. Оценка модели. После завершения обучения модель была оценена на тестовом наборе данных с использованием метрики точности и функции потерь.

5. Сохранение модели: Итоговая модель была сохранена в формате .h5 для последующего использования.

Оценку работы сети можно увидеть в таблице 1.

Таблица 1 – Оценка работы модели

	precision	recall	fl–score	support
with_mask	0.99	0.86	0.92	383
without_mask	0.88	0.99	0.93	384
Accuracy			0.93	767
Marco avg	0.93	0.93	0.93	767
Weighted avg	0.93	0.93	0.93	767

Выборочные результаты детекции представлены на рисунке 5.

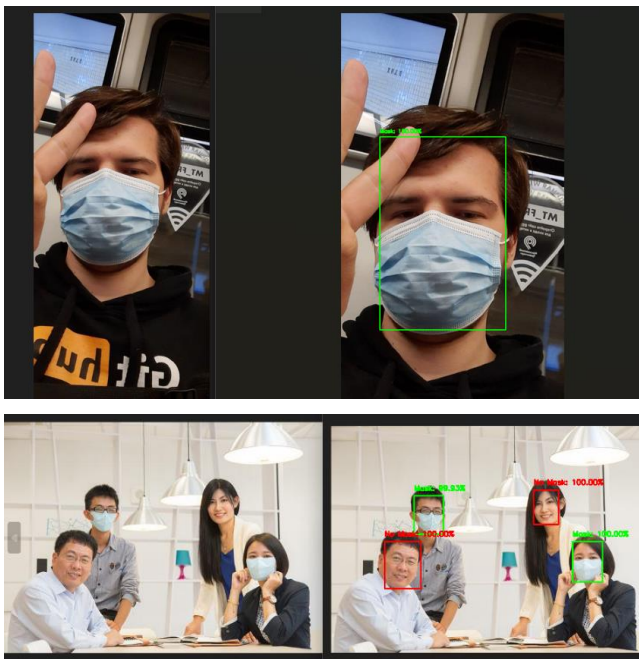


Рисунок 5 – Пример работы нейросети

E. YOLOv7

YOLOv7 представляет собой одну из современных и высокоэффективных архитектур для задач детектирования объектов в реальном времени. Использование этой модели позволяет добиться высокой точности и скорости распознавания, что особенно важно для приложений в области здравоохранения и общественной безопасности.

Первым этапом работы является подготовка данных. В качестве исходных данных используются изображения и соответствующие аннотации в формате XML. Для преобразования аннотаций в формат, совместимый с YOLO, используется следующая процедура:

1. Чтение аннотаций: Исходные аннотации в формате XML считываются и обрабатываются с помощью библиотеки xml.etree.ElementTree.

2. Преобразование координат: Координаты ограничивающих рамок (bounding boxes), указанные в аннотациях, преобразуются в формат YOLO, который использует нормализованные значения координат центра объекта, а также его ширины и высоты относительно размеров изображения.

YOLOv7, как и предыдущие версии YOLO, представляет собой одноэтапный детектор объектов, который выполняет детектирование и классификацию объектов в одном проходе. Основные компоненты архитектуры YOLOv7 включают:

1. Backbone. Основная часть сети, состоящая из сверточных слоев, используется для извлечения признаков из входного изображения. В YOLOv7 используется улучшенная версия CSPDarknet, которая оптимизирована для повышения точности и скорости работы;

2. Neck. Эта часть сети включает дополнительные сверточные слои и слои объединения признаков (feature pyramid networks), которые позволяют объединить признаки разных уровней абстракции [4]. Это улучшает способность модели обнаруживать объекты разного размера;

3. Head. В выходной части сети располагаются сверточные слои, которые выполняют предсказание координат ограничивающих рамок, классов объектов и вероятностей наличия объектов в различных ячейках сетки [5].

Архитектура YOLOv7 схематично представлена на рисунке 6

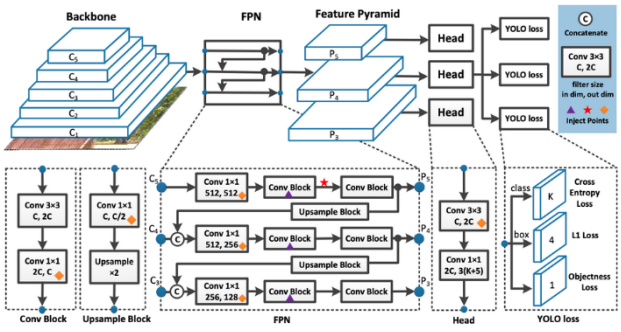


Рисунок 6 – Архитектура YOLOv7

Для обучения модели YOLOv7 используется набор данных, содержащий изображения лиц людей, с аннотациями наличия или отсутствия медицинских масок. Обучение выполняется с использованием оптимизатора Adam и функции потерь, учитывающей как точность предсказания координат ограничивающих рамок, так и точность классификации объектов.

После завершения обучения модель оценивается на тестовом наборе данных. Основные метрики, используемые для оценки, включают точность (accuracy), полноту (recall), и F1-меру. Эти метрики позволяют объективно оценить эффективность модели в задаче детектирования медицинских масок на лицах.

Использование модели YOLOv7 для задачи обнаружения медицинских масок на лицах демонстрирует высокую эффективность и производительность. Применение современных архитектур глубокого обучения позволяет решать задачи детектирования объектов в реальном времени, что особенно важно для приложений в области здравоохранения и общественной безопасности.

Результат работы сети представлен на рисунке 7.

Рисунок 7 – Результат детекции модели YOLOv7

III. Сравнение

Для анализа эффективности различных архитектур нейронных сетей в задаче распознавания медицинских масок были выбраны две модели: MobileNetV2 и YOLOv7. Обе модели обладают своими уникальными достоинствами и недостатками, что необходимо учитывать при выборе подхода для конкретного применения.

MobileNetV2 представляет собой легкую архитектуру сверточной нейронной сети, разработанную для использования в условиях ограниченных вычислительных ресурсов. Ее ключевые преимущества включают компактность и высокую скорость работы, что делает модель подходящей для применения на мобильных устройствах и в реальных временных системах.

Однако, несмотря на эти положительные аспекты, MobileNetV2 сталкивается с трудностями при распознавании медицинских масок на изображениях, где присутствует большое количество людей. Основные проблемы заключаются в:

1. Ограниченной способности к детекции множества объектов на одном изображении.

2. Снижении точности при увеличении плотности объектов в кадре.

Таким образом, эффективность MobileNetV2 значительно падает в условиях, когда необходимо анализировать сцены с множеством лиц, что ограничивает применимость модели в таких сценариях.

YOLOv7 демонстрирует высокую производительность в задачах детекции объектов в реальном времени.

Основное преимущество этой модели заключается в ее способности эффективно обрабатывать изображения с большим количеством лиц и точно определять наличие масок на них. Это достигается за счет:

1. Продвинутой архитектуры, оптимизированной для обработки сложных сцен.

2. Высокой точности и скорости работы, что позволяет применять модель в реальных условиях с большими объемами данных.

В отличие от MobileNetV2, YOLOv7 справляется с задачей распознавания масок на изображениях с множеством людей, обеспечивая высокую точность и надежность детекции. Это делает YOLOv7 предпочтительным выбором для применения в системе контроля ношения медицинских масок в общественных местах и других подобных сценариях.

IV. Заключение

В заключение, проведенное сравнение моделей MobileNetV2 и YOLOv7 в задаче распознавания медицинских масок показывает, что каждая из них обладает своими уникальными особенностями и областями применения. MobileNetV2 эффективна в условиях ограниченных ресурсов и низкой плотности объектов на изображении, в то время как YOLOv7 демонстрирует высокую производительность в сложных сценах с большим количеством лиц. Выбор модели должен основываться на конкретных требованиях и условиях эксплуатации системы распознавания медицинских масок.

Литература

[1] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-CostNavigation System for UAV," 2023 30th Saint Petersburg InternationalConference on Integrated Navigation Systems (ICINS), SaintPetersburg, Russian Federation, 2023, pp. 1-6, doi:10.23919/ICINS51816.2023.10168469.

[2] Ali, B., Sadekov, R.N., Tsodokova, V.V., A Review of NavigationAlgorithms for Unmanned Aerial Vehicles Based on Computer VisionSystems, Gyroscopy and NavigationЭта ссылка отключена., 2022

[3] Guzhva, N.S., Prun, V.E., Postnikov, V.V., Sadekov, R.N., Sholomov,D.L., Using 3D Object Detection DNN in an Autonomous Tram toPredict the Behaviour of Vehicles in the Road Scene, 29th SaintPetersburg International Conference on Integrated Navigation Systems,ICINS 2022

[4] Guzhva, N.S., Ali, B., Bakulev, K.S., Sadekov, R.N., Sholokhov, A.V.Evaluating the Accuracy of Tram Positioning System in High-RiseBuilding Environment Using Data from Visual GeoinformationSystems, 30th Anniversary Saint Petersburg International Conferenceon Integrated Navigation Systems, ICINS 2023, 2023

[5] Chollet, F. "Deep Learning with Python." Manning Publications, pp. 120-135, 2018.

[6] Abadi, M., et al. "TensorFlow: A System for Large-Scale Machine Learning." Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, pp. 265-283, 2016.

[7] Howard, A.G., et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv preprint arXiv:1704.04861, pp. 80-97, 2017.

[8] Ivanov A. V., Petrov S. N. "Specialized Algorithms for Object Detection: Analysis and Application of YOLOv7". Journal of Computer Sciences, 2022, pp. 45-58..

ГРНТИ: 28.23.15
ASJC: 1707
OECD: 01.02.EP

Детекция беспилотных летательных аппаратов на фотографиях с использованием методов компьютерного зрения

Д. С. Матяш
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m1910814@edu.misis.ru

***Аннотация* — на протяжении последних десятилетий беспилотные летательные обороты, чаще именуемые как дроны, претерпевали кардинальные изменения, которые значительно улучшили их структуру, летные характеристики, методологию работы и контроль, с точки зрения навигации. Все перечисленное привело к заметной экспансии области применения данных технологий. Быстрое распространение беспилотных летательных аппаратов (БПЛА) требует разработки надежных и эффективных систем обнаружения. Такие системы имеют решающее значение для обеспечения безопасности в различных областях: безопасность аэропортов и воздушного пространства, личная безопасность и конфиденциальность, а также защита критически важной инфраструктуры. В данной статье представлено комплексное исследование по обнаружению беспилотников на фотографиях с использованием передовых методов компьютерного зрения.**
В статье приводится подробный анализ эффективности различных нейросетевых архитектур на крупном наборе данных изображений, полученных путем съемки беспилотных летательных аппаратов.

Ключевые слова — детекция объектов, бпла, uav, YOLO, CNN, ResNet, VGG.

I. Введение

В последние годы дроны стали повсеместно использоваться в различных сферах, таких как аэрофотосъемка, наблюдение и доставка грузов.На Scopus и других подобных web-ресурсах можно найти тысячи статей с использованием терминов “unmanned aerial vehicle” или “uav”, что говорит о высоком интересе к теме со стороны научного сообщества. Крупные влиятельные компании, а также ряд вооруженных сил разных стран вкладывают миллиарды долларов в рынок БПЛА [1], [2].

Однако их нерегулируемое использование может представлять опасность для частной жизни, безопасности и общественного порядка. В связи с этим растет потребность в эффективных методах обнаружения и идентификации дронов на изображениях и видео [3].

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 [4], [5].

Цель данного исследования - изучить возможности применения сверточных нейронных сетей (CNN) для обнаружения дронов на фотографиях.

II. Набор данных

Основой обучающего набора данных стал набор данных Drone Dataset (UAV), который содержит 1359 изображений БПЛА и все они помечены. Представлены файлы различных разрешений для обучения на разных моделях, таких как YOLO, Tensorflow и PyTorch [6]. В набор данных включены только БПЛА с винтами (квадрокоптеры, трикоптеры и тд.). Другие типы дронов в датасете не представлены. Примеры изображений показаны на Рисунке 1.

Рисунок 1 – Пример изображений из датасета Drone Dataset (UAV)

III. НЕЙРОСЕТЕВЫЕ АРХИТЕКТУРЫ

Нейронные сети - это мощные модели глубокого обучения, которые произвели революцию в различных областях, включая обнаружение объектов. Они способны идентифицировать и определять местоположение объектов на изображениях или видео с поразительной точностью.

Одной из наиболее распространенных задач обнаружения, для решения которых используются нейронные сети, является обнаружение объектов на изображениях. Обучив нейронную сеть на наборах помеченных изображений с интересующими объектами, сеть может научиться обнаруживать и локализовать эти объекты на новых изображениях.

Сети обнаружения объектов обычно состоят из опорной сети, которая извлекает признаки из входных данных, и detection head, которая предсказывает наличие и местоположение объектов. Схематично работа такой сети показана на Рисунке 2. Опорная сеть часто представляет собой предварительно обученную сверточную нейронную сеть (CNN), такую как ResNet или VGGNet, которая изучила общие характеристики изображения. Detection head обычно представляет собой сеть предложений регионов (RPN) или детектор одиночных снимков (SSD).

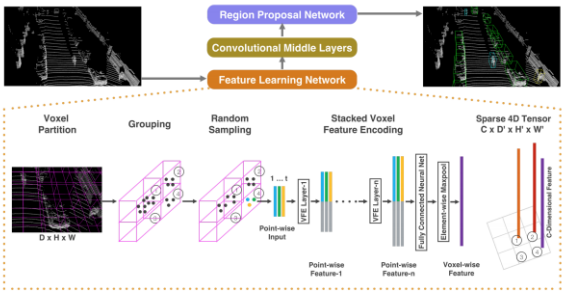


Рисунок 2 – Сеть обнаружения объектов

RPN генерируют регионы-кандидаты на объекты, известные как области интереса (ROI), и классифицируют их как содержащие или не содержащие объекты. SSD, с другой стороны, напрямую предсказывают границы объектов и вероятности классов для каждого пикселя входного изображения.

После обнаружения и локализации объектов часто используются дополнительные шаги постобработки, такие как подавление без максимума, для удаления дублирующих или перекрывающих друг друга обнаружений и уточнения границ.

Одним из популярных типов нейросетевых архитектур, используемых для обнаружения объектов, является нейросеть с остаточными функциями (ResNet). ResNet известна своей глубокой архитектурой с сотнями слоев.

Еще одна широко используемая архитектура нейронной сети для обнаружения объектов - сеть Visual Geometry Group Network (VGG). VGG известна своей простотой и однородной структурой, включающей несколько конволюционных слоев и максимальные слои объединения.

Yolo (You Only Look Once) - еще одна популярная нейросетевая архитектура для обнаружения объектов. Yolo известна своей скоростью и точностью, поскольку она обрабатывает все изображение сразу и выводит ограничительные рамки и метки классов для обнаруженных объектов в режиме реального времени.

В целом, такие нейронные сети, как ResNet, VGG и Yolo, являются мощными инструментами для задач обнаружения объектов, поэтому данные модели были выбраны для исследования.

A. VGG-16

Модель VGG-16 - это глубокая конволюционная нейронная сеть, предназначенная для задач классификации изображений. Она была представлена исследователями из Visual Geometry Group в Оксфордском университете в 2014 году. VGG-16 - это вариант оригинальной модели VGG, которая была разработана для ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Модель VGG-16 широко используется в исследованиях и приложениях компьютерного зрения благодаря своей высокой точности и простоте использования. Она стала эталонной моделью для оценки производительности других моделей глубокого обучения в задачах классификации изображений. Одной из причин ее популярности являются результаты работы на наборе данных ImageNet, где она достигла уровня ошибок в топ-5 ниже, чем предыдущие современные модели (Рисунок 3).

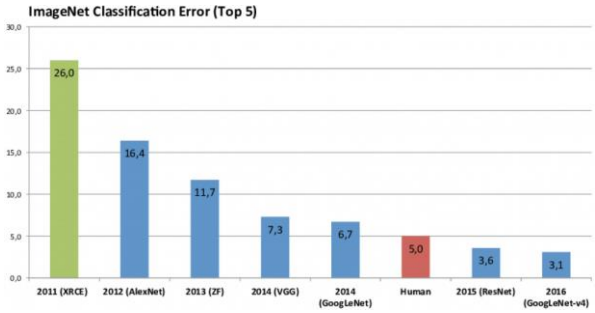


Рисунок 3 – Ошибка при классификации ImageNet

Одной из ключевых особенностей VGG-16 является ее простота и унифицированная архитектура. Модель состоит из 16 слоев, включая 13 конволюционных слоев и 3 полностью связанных слоя. За каждым конволюционным слоем следует слой max pooling, который помогает уменьшить пространственные размеры входных данных и извлечь важные особенности. Такая архитектура позволяет VGG-16 достигать отличной производительности в задачах классификации изображений, особенно при обучении на больших наборах данных, таких как ImageNet.

Эта конструкция характеризуется использованием компактных конволюционных фильтров 3×3 и глубоких архитектур с размером страйда 1. Объединяющие слои имеют конфигурацию 2×2 с размером страйда 2 и сохраняют ту же прокладку. По умолчанию сеть VGG-16 обрабатывает входные изображения размером 224×224. Перед полностью связанными слоями используется карта признаков 7×7, содержащая 512 каналов, которая затем преобразуется в вектор с 25 088 каналами (7×7×512) в качестве результирующего представления

признаков. На рисунке 4 приведена структура сети VGG-16.

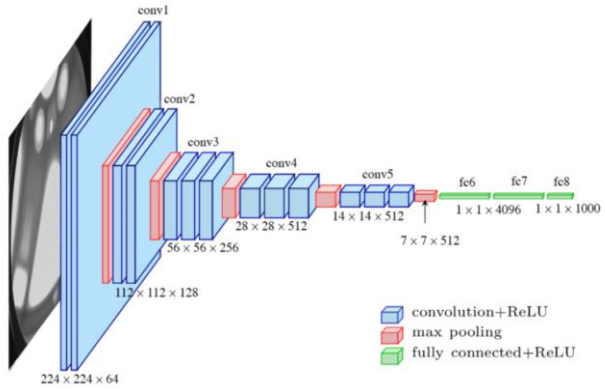


Рисунок 4 – Схематичное изображение структуры VGG-16

Несмотря на впечатляющую производительность, VGG-16 имеет ряд ограничений. Одним из главных недостатков модели является большое количество параметров, что может сделать обучение и вывод результатов медленным и вычислительно дорогим. Кроме того, унифицированная архитектура VGG-16 может быть не оптимальной для всех типов задач классификации изображений, поскольку некоторые наборы данных могут требовать более специализированных архитектур для достижения оптимальной производительности.

B. ResNet50

ResNet50 - это модель глубокой нейронной сети, получившая популярность благодаря высокой точности и эффективности в задачах распознавания изображений. Разработанная исследователями Microsoft Research в 2015 году, ResNet50 основана на фреймворке остаточного обучения, который позволяет с легкостью обучать очень глубокие сверточные нейронные сети.

Одним из ключевых преимуществ ResNet50 является его глубокая архитектура, состоящая из 50 слоев (Рисунок 5), что позволяет ей изучать сложные особенности изображений. Такая глубина необходима для того, чтобы учесть сложность реальных наборов данных и достичь высочайшей производительности в таких задачах, как классификация изображений, обнаружение объектов и сегментация изображений.

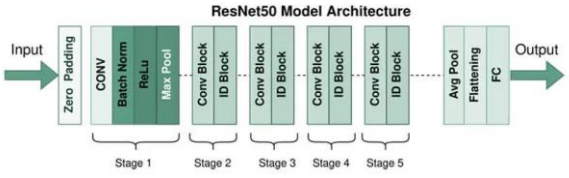


Рисунок 5 – Схематичное изображение архитектуры ResNet50

Еще одна примечательная особенность ResNet50 - использование соединений быстрого доступа, которые позволяют сети обходить определенные слои во время обучения. Это помогает смягчить проблему исчезающего градиента, обычно встречающуюся в очень глубоких нейронных сетях, и обеспечивает более быструю сходимость в процессе обучения. В результате ResNet50 можно обучать более эффективно и с меньшим количеством параметров по сравнению с другими сетевыми архитектурами.

Что касается производительности, то ResNet50 была тщательно протестирована на стандартных наборах данных изображений, таких как ImageNet, и неизменно показывала самые высокие результаты в задачах классификации изображений. С впечатляющим показателем ошибок в топ-5 около 3,6 % (Рисунок 3) ResNet50 превосходит многие другие модели глубоких нейронных сетей и демонстрирует превосходные возможности в распознавании и классификации широкого спектра изображений.

Кроме того, было показано, что ResNet50 хорошо обобщает данные, не встречающиеся в поле зрения, и демонстрирует хорошую устойчивость к изменениям освещения, фона и качества изображения. Это очень важно для применения модели в реальных приложениях, где изображения могут поступать из разных источников и иметь различные характеристики.

Несмотря на свои достоинства, ResNet50 может оказаться не лучшим выбором для всех задач распознавания изображений. Его большой размер и вычислительная сложность могут сделать обучение и развертывание модели ресурсоемким, особенно на оборудовании с ограниченными возможностями памяти и обработки данных. Кроме того, тонкая настройка и адаптация ResNet50 для специализированных задач может потребовать дополнительных знаний и усилий.

В заключение можно сказать, что ResNet50 - это мощная модель глубокой нейронной сети, которая отлично справляется с задачами распознавания изображений, особенно в сценариях, где важны высокая точность и устойчивость. Глубокая архитектура, эффективный процесс обучения и впечатляющая производительность на эталонных наборах данных делают ее популярной среди исследователей и практиков, работающих в области компьютерного зрения. Несмотря на то, что ResNet50 может подойти не для всех приложений, его возможности и универсальность делают его ценным инструментом для развития передовых технологий обработки и анализа изображений.

C. YOLO

Среди различных алгоритмов обнаружения объектов выделяется система YOLO (You Only Look Once), которая отличается замечательным балансом скорости и точности, позволяя быстро и надежно идентифицировать объекты на изображениях. С момента своего появления семейство YOLO прошло через множество итераций, каждая из которых развивала предыдущие версии, устраняя ограничения и повышая производительность (Рисунок 6) [7].

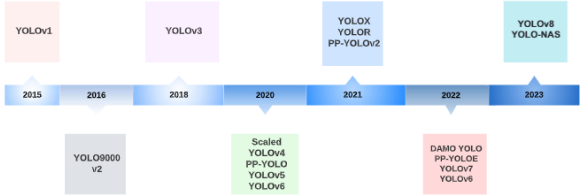


Рисунок 6 – Версии YOLO на временной прямой

YOLO делит входное изображение на равные ячейки сеткой $S \times S$. Каждая ячейка сетки отвечает за предсказание объекта и предсказывает B ограничивающих рамок и соответствующие им оценки уверенности (вероятности).

После получения ограничивающих рамок производится подавление немаксимумов (non-maximum suppression, NMS), которое позволяет избавиться от нерелевантных рамок. Это необходимо, потому что нейронная сеть часто находит несколько рамок, в которые попадает один и тот же объект (Рисунок 7) [8].

Рисунок 7 – Фильтрация нерелевантных ограничивающих рамок при помощи NMS

YOLOv8 была выпущена в январе 2023 года компанией Ultralytics, которая разработала YOLOv5. В YOLOv8 было представлено пять масштабных версий: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) и YOLOv8x (extra-large). YOLOv8 поддерживает множество задач технического зрения, таких как обнаружение объектов, сегментация, оценка положения, отслеживание и классификация [9].

На Рисунке 8 показана подробная архитектура YOLOv8. 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.

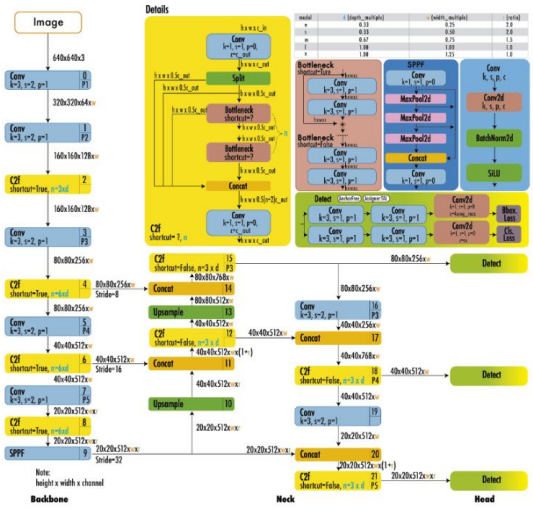


Рисунок 8 – Схематичное изображение архитектуры ИНС YOLOv8

В YOLOv8 используется безъякорная модель с разделенной head для независимой обработки задач объектности, классификации и регрессии [10]. Такая конструкция позволяет каждой ветви сосредоточиться на своей задаче и повышает общую точность модели. В выходном слое YOLOv8 в качестве функции активации для оценки объектности используется сигмоидная функция, представляющая собой вероятность того, что в ограничительной рамке находится объект. В слое YOLOv8 используется функция softmax для вероятностей классов, представляющих вероятность принадлежности объектов к каждому возможному классу. В слое YOLOv8 используются функции потерь CIoU и DFL для потерь при определении границ и двоичная кросс-энтропия для потерь при классификации. Эти потери улучшили эффективность обнаружения объектов, в основном при работе с небольшими объектами.

Слои сети объединяются в блоки:

- Backbone – последовательность из блоков Conv и C2f, которые выполняют функции свертки и находят карты признаков, с пирамидой масштабов, которая используется для объединения признаков;
- Conv, C2f (cross-stage partial bottleneck with two convolutions) – являются составными частями сети Backbone;
- SPPF (spatial pyramid pooling fast) – пирамида масштабов, используемая для объединения признаков на разных масштабах;
- Upsample – слой повышения размерности карты признаков, используется для согласования входных размеров;
- Head – финальная последовательность слоев для формирования ограничивающих рамок и классов объектов при помощи блоков Detect;
- Detect – ряд слоев с использованием свертки с размером ядра 1, что позволяет уменьшать (или увеличивать) количество каналов. Подобные слои часто используются в полностью сверточных нейронных сетях в последних слоях сети.

YOLOv8 можно запустить из интерфейса командной строки (CLI), а также установить в виде пакета PIP. Кроме того, в комплект поставки входит множество интеграций для маркировки, обучения и развертывания. При оценке на наборе данных MS COCO test-dev 2017 YOLOv8x достиг AP 53,9% при размере изображения 640 пикселей (по сравнению с 50,7% у YOLOv5 при том же размере входных данных) со скоростью 280 кадров в секунду на NVIDIA A100 и TensorRT [11].

D. Описание метрик

Recall, precision и accuracy - три распространенные метрики, используемые для оценки эффективности моделей классификации. Каждая метрика помогает понять различные аспекты работы модели с точки зрения правильной идентификации и классификации экземпляров [12].

Для оценки детекции объектов используется вспомогательная метрика IoU (intersection over union) для сравнения двух ограничивающих рамок, которая вычисляется как отношение площади пересечения к площади их объединения (Рисунок 9).

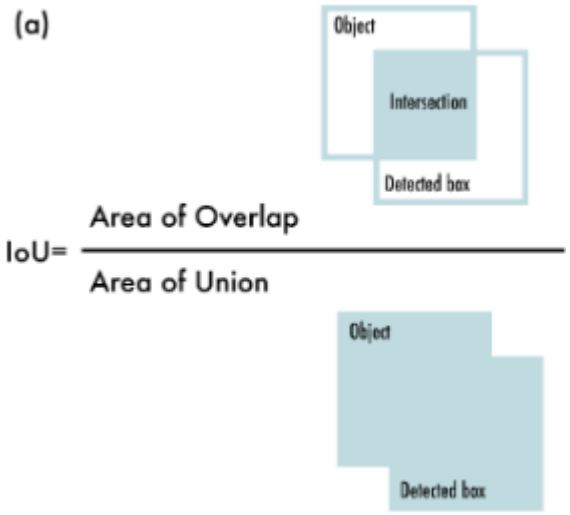


Рисунок 9 – Визуализация метрики IoU

Метрика IoU лежит в диапазоне [0, 1] и чем больше ее значение, тем сильнее совпадают ограничивающие рамки [7]. С ее помощью можно определить метрики Precision и Recall. Метрики задаются для двух множеств ограничивающих рамок $y = y_i$ и $\hat{y} = \hat{y}_i$. Каждая ограничивающая рамка содержит в себе индекс изображения, индекс класса, вероятность класса, координаты ограничивающей рамки. После детекции все рамки распределяются на 3 категории:

- предсказание считается True Positive когда метрика IoU для конкретной предсказанной рамки больше или равна некоторому пороговому значению;
- предсказание считается Flase Positive когда IoU меньше некоторого порогового значения, или является дубликатом (эталонная рамка уже была соотнесена с другой предсказанной рамкой);
- предсказание считается False Negative когда-либо на изображении с эталонной рамкой определенного класса не было найдено ни одного объекта (считается факт отсутствия предсказания).

Precision измеряет долю правильно идентифицированных экземпляров среди всех экземпляров, которые модель классифицировала как принадлежащие к данному классу. Она рассчитывается как отношение истинно положительных результатов к сумме истинно положительных и ложноположительных результатов (экземпляров, неверно классифицированных как принадлежащие данному классу). Высокий показатель точности указывает на то, что модель точно классифицирует экземпляры, и в нее попадает меньше ложных срабатываний [13].

$$precision = \frac{TP(c)}{TP(c) + FP(c)} \quad (1)$$

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

Recall, также известная как чувствительность, - это мера того, насколько хорошо модель может правильно идентифицировать экземпляры определенного класса. Она рассчитывается как отношение истинно положительных результатов (правильно идентифицированных экземпляров) к сумме истинно положительных и ложноотрицательных результатов (неправильно идентифицированных экземпляров). Высокий показатель recall указывает на то, что модель хорошо улавливает все экземпляры класса, при этом меньше экземпляров пропускается.

$$recall = \frac{TP(c)}{TP(c) + FN(c)} \quad (2)$$

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

Accuracy - это показатель того, насколько хорошо модель может правильно классифицировать экземпляры по всем классам. Она рассчитывается как отношение суммы истинно положительных и истинно отрицательных результатов (правильно классифицированных экземпляров) к общему количеству экземпляров. Точность дает общее представление о производительности модели, но может быть не столь информативной при работе с несбалансированными наборами данных, где один класс доминирует в распределении экземпляров.

$$accuracy = \frac{\text{Кол} - \text{во правильных предсказаний}}{\text{Общее количество изображений на вход}} \quad (3)$$

В целом, recall нацелена на выявление всех экземпляров определенного класса, precision - на точную классификацию экземпляров как принадлежащих к определенному классу, а accuracy - на общую оценку эффективности модели по всем классам. Все три показателя важны для оценки эффективности модели классификации и могут помочь определить области, в которых модель может нуждаться в улучшении

IV. СРАВНЕНИЕ

Были обучены 3 модели VGG-16, ResNet50 и YOLO small.

Первой была обучена VGG-16 на картинках размером 256×256, batch_size=16, epoch - 50.

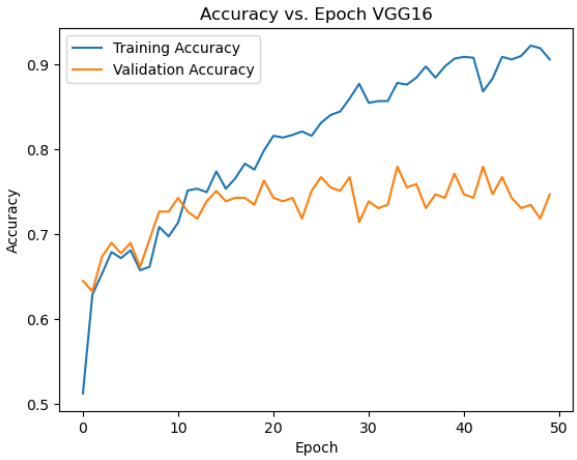


Рисунок 10 – График accuracy в процессе обучения на тренировочных и тестовых данных VGG-16

Финальные результаты модели после 50 эпох обучения представлены в Таблице 1.

ТАБЛИЦА 1 МЕТРИКИ VGG-16

Тренировочные Данные	Accuracy	0.90594
	Precision	0.88443
	Recall	0.90643
Тестовые Данные	Accuracy	0.78694
	Precision	0.77451
	Recall	0.79121

Также на Рисунке 11 можно увидеть график Loss функции по эпохам.

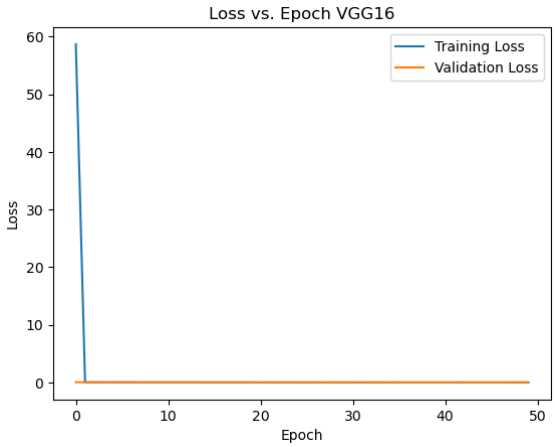


Рисунок 11 – График loss функции в процессе обучения на тренировочных и тестовых данных VGG-16

Второй на очереди, стала модель ResNet50. Обучение проводилось на картинках размером 256×256, batch_size=16, epoch - 50.

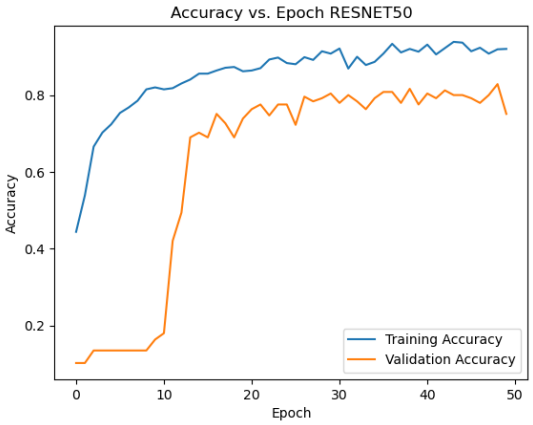


Рисунок 12 – График accuracy в процессе обучения на тренировочных и тестовых данных ResNet50

Финальные результаты модели после 50 эпох обучения представлены в Таблице 2.

ТАБЛИЦА 2 МЕТРИКИ RESNET50

Тренировочные Данные	Accuracy	0.92025
	Precision	0.92447
	Recall	0.91641
Тестовые Данные	Accuracy	0.79102
	Precision	0.79453
	Recall	0.79022

Также на Рисунке 13 можно увидеть график Loss функции по эпохам.

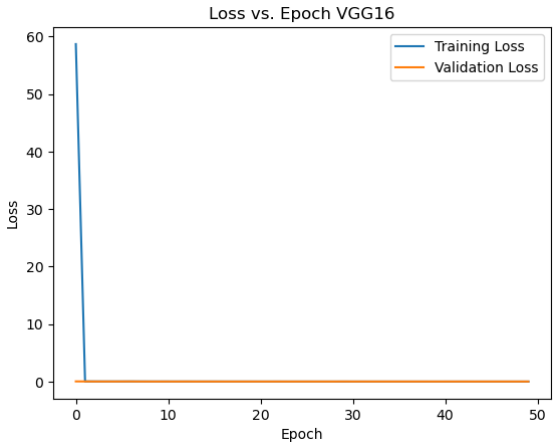


Рисунок 13 – График loss функции в процессе обучения на тренировочных и тестовых данных ResNet50

И последней моделью была YOLOv 8s. Обучение проводилось на картинках размером 256×256, batch_size=8, epoch - 30.

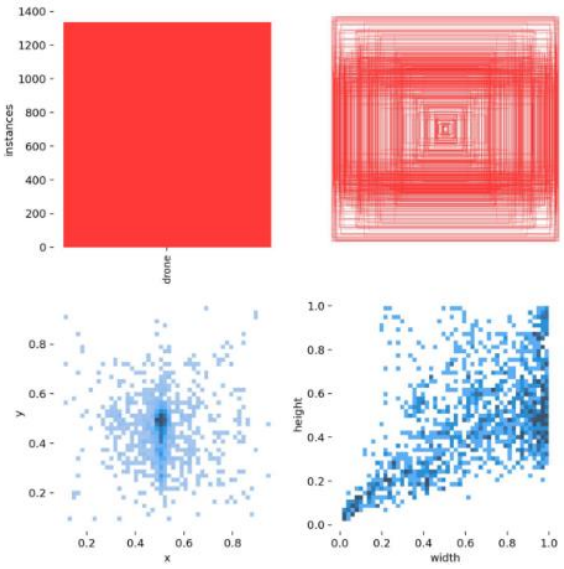


Рисунок 14 – Распределение баундинг боксов и классов

На Рисунке 15 показан график метрики Precision для YOLOv8 в процессе обучения.

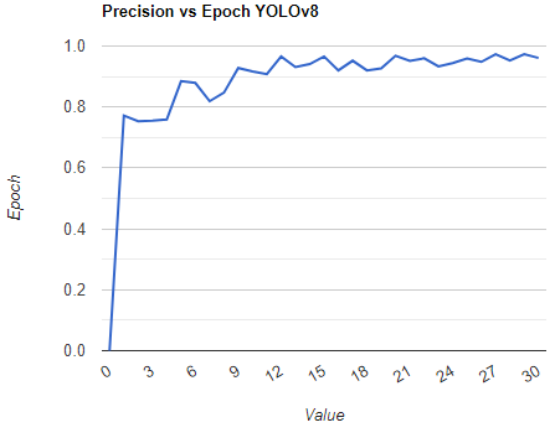


Рисунок 15 – График График precision в процессе обучения YOLOv8

Финальные результаты модели после 30 эпох обучения представлены в Таблице 3.

ТАБЛИЦА 3 МЕТРИКИ YOLOV8

Тренировочные Данные	Accuracy	0.953
	Precision	0.953
	Recall	0.954
Тестовые Данные	Accuracy	0.824
	Precision	0.824
	Recall	0.825

На рисунке 16 представлен пример успешной детекции дрона на собственной тестовой картинке.

Рисунок 16 – Пример детекции дронов YOLOv8

Однако, конечно, были и изображения на которых БПЛА не был обнаружен, например, из-за необычного положения, малого размера, перекрытия другими объектами и тд, пример такого изображения ниже. Соответственно, можно сделать вывод, что данные технологии подходят, для обнаружения четко распознаваемых бпла на изображениях или видео.

Рисунок 17 – Пример изображения, где не удалось обнаружить дрон

В таблице 4 приведены значения метрик для всех трех рассматриваемых на тестовых данных нейронных сетей.

ТАБЛИЦА 4 СВОДНАЯ ТАБЛИЦА МЕТРИК

	YOLOv8	ResNet50	VGG-16
Accuracy	0.824	0.79102	0.78694
Precision	0.824	0.79453	0.77451
Recall	0.825	0.79022	0.79121

Все три модели показали неплохие результаты, однако YOLOv8 проявила себя чуть лучше, нежели весьма близки по значениям друг к другу ResNet50 и VGG-16. Стоит напомнить, что датасет составлен только из хорошо видимых человеческому глазу дронов с винтовой конструкцией.

V. Заключение

В данной работе были изучены методы обнаружения дронов на изображениях при помощи нейронных сетей. В частности, были описаны, обучены и протестированы архитектуры YOLOv8, ResNet50 и VGG-16 на наборе данных Drone Dataset (UAV). Были сравнены 3 модели и описаны метрики, используемые в задачах обнаружения объектов.

Анализ метрик показал, что максимальная точность достигается YOLOv8. Вот почему YOLOv8 может оказаться лучше в области обнаружения дронов:

- Производительность в реальном времени: YOLOv8 создан для скорости и эффективности, что делает его идеальным для приложений реального времени, таких как обнаружение дронов, где важны быстрые результаты.
- Локализация объектов: YOLOv8 фокусируется на точном определении местоположения и границ объектов на изображениях, что делает его идеальным для идентификации и отслеживания беспилотников.
- Эффективная архитектура: YOLOv8 использует легкую и оптимизированную архитектуру, требующую меньше вычислительной мощности, чем VGG-16 и ResNet50.

VGG-16 и ResNet50:

- Мастерство классификации изображений: Эти модели отлично справляются с классификацией изображений по определенным категориям, но по своей сути они не предназначены для обнаружения объектов.
- Сильные стороны глубокого обучения: VGG-16 и ResNet50 сложны и требуют больших вычислительных затрат, что делает их более подходящими для задач, где точность имеет первостепенное значение.

Почему YOLOv8 может превзойти их в обнаружении дронов:

1. Меньшие требования к набору данных: Наборы данных для обнаружения дронов обычно меньше, чем общие наборы данных изображений. Эффективность YOLOv8 позволяет ему добиваться хороших результатов даже при ограниченном объеме данных.
2. Фокусировка на границах объектов: Обнаружение дронов требует точной локализации. Архитектура YOLOv8 специально разработана для определения границ объектов, что повышает ее способность точно идентифицировать беспилотники.
3. Адаптация к различным размерам и положениям дронов: Гибкость YOLOv8 позволяет обучать ее на дронах различных размеров, что делает ее более универсальной, чем VGG-16 и ResNet50.

Литература

[1] Savon, D.Yu & Safronov, A.E. & Vikhrova, N.O. & Kruzhkova, Galina & Goncharov, M.S.. (2022). IMPACT OF THE CRISIS ON THE FINANCIAL PERFORMANCE OF THE COAL INDUSTRY. Ugol'. 62-68. 10.18796/0041-5790-2022-11-62-68.

[2] F. Nex, C. Armenakis, M. Cramer, D.A. Cucci, M. Gerke, E. Honkavaara, A. Kukko, C. Persello, J. Skaloud, UAV in the advent of the twenties: Where we stand and what is next, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 184, 2022, Pages 215-242, ISSN 0924-2716, https://doi.org/10.1016/j.isprsjprs.2021.12.006

[3] Gregory M. Crutsinger, Jason Short, and Roger Sollenberger. The future of UAVs in ecology: an insider perspective from the Silicon Valley drone industry. Journal of Unmanned Vehicle Systems. 4(3): 161-168. https://doi.org/10.1139/juvs-2016-0008

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

[5] Ahmed, F., Mohanta, J.C., Keshari, A. et al. Recent Advances in Unmanned Aerial Vehicles: A Review. Arab J Sci Eng 47, 7963–7984 (2022). https://doi.org/10.1007/s13369-022-06738-0

[6] Ahmed, F., Mohanta, J.C., Keshari, A. et al. Recent Advances in Unmanned Aerial Vehicles: A Review. Arab J Sci Eng 47, 7963–7984 (2022). https://doi.org/10.1007/s13369-022-06738-0

[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] “A Comprehensive Review of YOLO: From YOLOv1 and Beyond” available at https://arxiv.org/pdf/2304.00501.pdf (Accessed December 19, 2023).

[9] Исследование возможности распознавания объектов на спутниковых снимках. Available from: https://www.researchgate.net/publication/376809371_Issledovanie_vozmoznosti_raspoznavania_obektov_na_sputnikovyh_snimkah

[10] “Ultralytics YOLOv8” available at https://github.com/ultralytics/ultralytics (Accessed December 19, 2023)

[11] Vijayakumar, Ajantha & Vairavasundaram, Subramaniyaswamy. (2024). YOLO-based Object Detection Models: A Review and its Applications. Multimedia Tools and Applications. 1-40. 10.1007/s11042-024-18872-y.

[12] Osipov, Aleksey & Shumaev, Vyacheslav & Ekielski, Adam & Gataullin, Timur & Suvorov, Stanislav & Mishurov, Sergey & Gataullin, Sergey. (2022). Identification and Classification of Mechanical Damage During Continuous Harvesting of Root Crops Using Computer Vision Methods. IEEE Access. 10. 1-1. 10.1109/ACCESS.2022.3157619.

[13] Smolin, Aleksandr & Yamaev, Andrei & Ingacheva, Anastasia & Shevtsova, Tatyana & Полевой, Дмитрий & Chukalina, Marina & Nikolaev, Dmitry & Arlazarov, Vladimir. (2022). Reprojection-Based Numerical Measure of Robustness for CT Reconstruction Neural Network Algorithms. Mathematics. 10. 4210. 10.3390/math10224210.

ГРНТИ коды: 20.01.07, 28.23.15, 28.23.37, 28.23.01
ASJC коды: 1707, 1702
OECD коды: 01.02.EP

Классификация катаракты глаза при помощи компьютерного зрения

М. Ф. Мельникова
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2300443@edu.misis.ru

***Аннотация*— катаракта является одним из наиболее распространённых заболеваний глаз, приводящим к снижению зрения и слепоте. В данной работе рассматриваются различные решения с использованием нейронных сетей для бинарной классификации зрелости катаракты на основе изображений глаз, то есть определения, является ли катаракта зрелой или незрелой. Цель данного исследования — выявление эффективных моделей, способных точно диагностировать степень зрелости катаракты. Анализ проведенных исследований не только предоставляет обзор различных методов классификации катаракты, но и подчеркивает сложности в поиске эффективных решений. Это исследование имеет высокую значимость в контексте развития современных методов медицинской диагностики и обогащения научных знаний в области офтальмологии.**

Ключевые слова — катаракта глаза, компьютерное зрение, нейронная сеть, CNN, Resnet50

I. Введение

Катаракта, являясь распространенным заболеванием глаз, приводит к ухудшению зрения и слепоте [1,2]. Точная и своевременная диагностика стадии катаракты имеет решающее значение для выбора оптимального метода лечения и сохранения зрения пациента.

В настоящее время диагностика катаракты основана на визуальном осмотре и использовании специализированных офтальмологических инструментов. Офтальмологи, тщательно изучая глаз пациента, выявляют визуальные признаки, такие как помутнение хрусталика, и определяют зрелость катаракты.

Этот процесс трудоемок, поскольку разные стадии катаракты могут иметь схожие симптомы и соответственно изображения глаза выглядят похоже, хотя стадии заболевания являются различными.

В связи с развитием современных методов медицинской диагностики существует острая необходимость в разработке автоматических, эффективных и точных методов определения стадии зрелости катаракты.

В последние годы глубокое обучение [3, 4] стало широко применяться в области классификации медицинских изображений, демонстрируя впечатляющие результаты. В рамках данной работы предлагается метод компьютерного зрения [5] для бинарной классификации зрелости катаракты.

В работе рассматриваются и сравниваются эффективность двух нейронных сетей [6] глубокого обучения: ResNet50 и VGG16, в задаче определения зрелости катаракты.

Для исследования были использованы два набора данных: один из открытых источников и один, собранный самостоятельно.

A. *Cataract Classification Dataset*

Набор общедоступных данных Cataract Classification Dataset [7], содержащий более 400 изображений 2 типов катаракты глаза. Набор данных содержит 2 класса, то есть это задача бинарной классификации. Первый класс – не зрелая катаракта. Второй класс – зрелая катаракта. (рисунок 1).

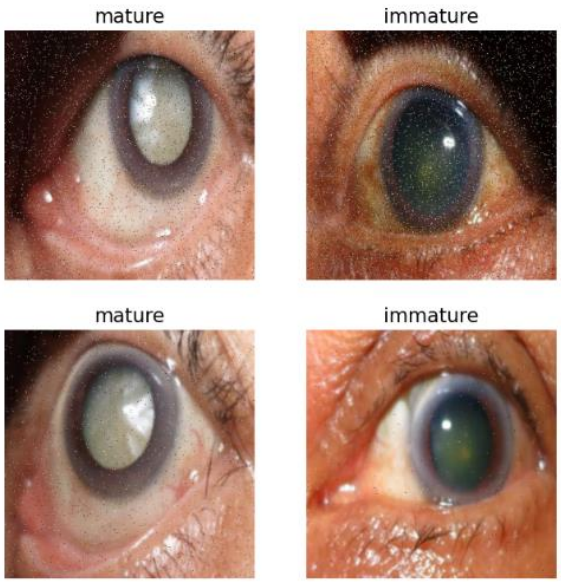


Рис. 1. Примеры изображений 2 классов

Для повышения устойчивости к различным условиям и характеристикам различных глаз, набор данных имеет разные цвета глаз, а также были фотографированы в разном ракурсе и приближении.

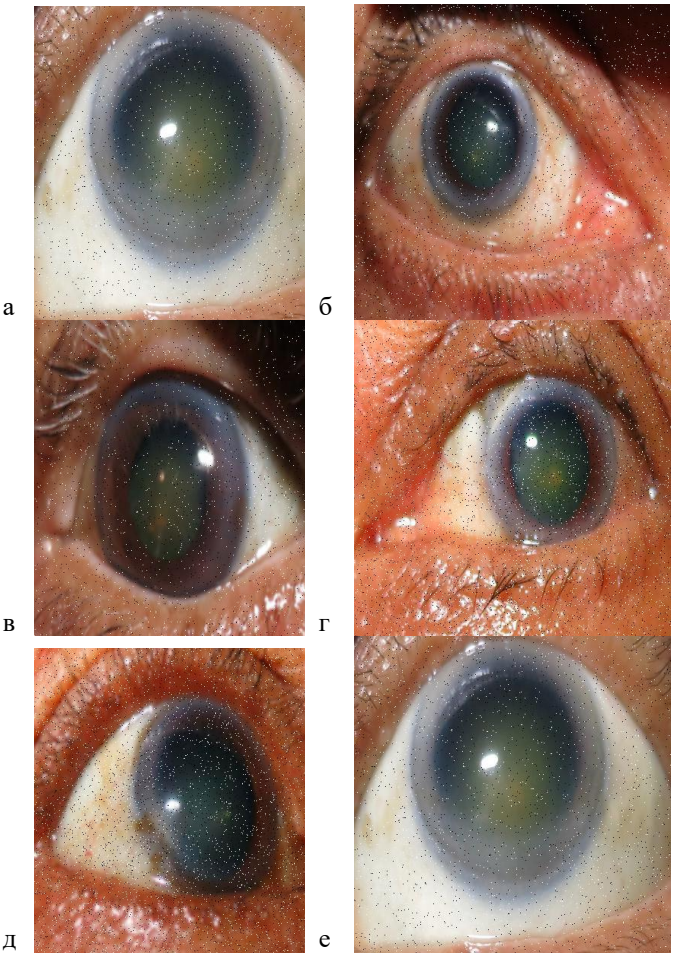


Рис. 2. а) приближенное изображение, б) отдаленное изображение, в) карие, г) голубые, д) смотрят в сторону, е) смотрят прямо

Интересно отметить, что некоторые изображения разных типов катаракты крайне тяжело отличить. Это отображено на рисунке 3.

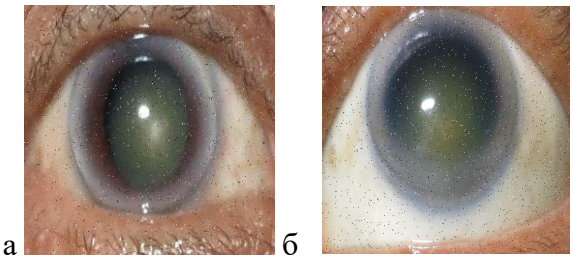


Рис. 3. Катаркты схожие внешне: а) не зрелая , б) зрелая катаракта

B. Собственный набор данных

Данный набор данных является дополненным, по отношению к набору данных «Cataract Classification Dataset». Набор данных «Cataract Classification Dataset» масштабный и охватывает полный объем необходимых данных для классификации катаракты, а дополнительные изображения необходимы для повышения качества проверки нейронных сетей. Примеры собственных изображений отражены на рисунке 4.

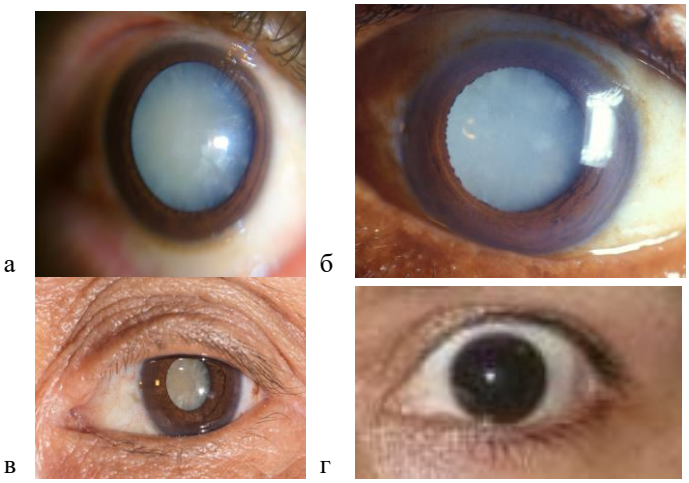


Рис. 4. Дополнительный набор данных а), б) зрелые катаракты, в), г) не зрелые

II. Нейросетевые архитектуры

В представленной работе для решения задачи классификации катаракты глаза использовались две нейронные сети: ResNet50[8] и VGG16[9]. Обе модели представляют собой мощные инструменты для извлечения высокоуровневых признаков из изображений, что важно для точной классификации сложных объектов, таких как катаракта глаза [10].

A. Resnet50

Архитектура модели

Модель для классификации катаракты основана на архитектуре ResNet50, предобученной на наборе данных ImageNet. Для нашей задачи модель была модифицирована путем замены последнего слоя, чтобы соответствовать количеству классов (2 класса: незрелая катаракта и зрелая катаракта). Схема архитектуры модели представлена на рисунке 5.

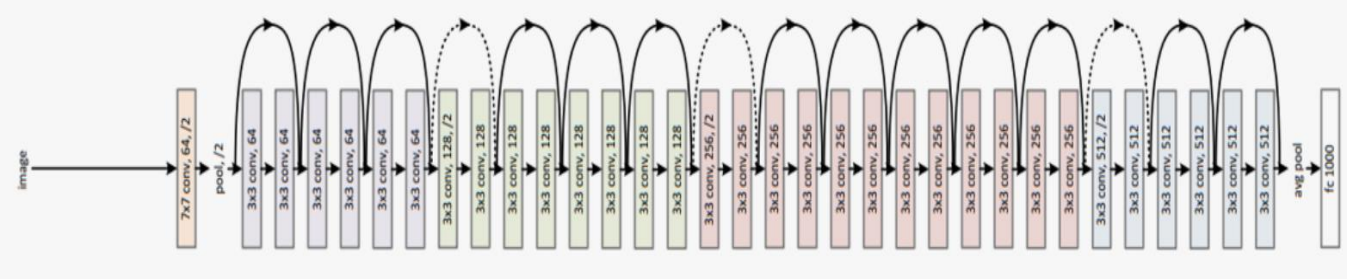


Рис. 5. Архитектура Resnet50

Входной слой (Input Layer):

Входные изображения размером 96×96×3

Скрытые слои (Hidden Layers):

Основная часть модели представляет собой архитектуру ResNet50, включающую серию остаточных (residual) блоков, которые позволяют строить глубокие нейронные сети без проблемы исчезающих градиентов.

ResNet50 состоит из следующих блоков:

1. начальные сверточные и максимальные объединяющие слои;
2. блоки остаточных слоев с разным количеством сверточных слоев в каждом блоке;
3. глобальный усредняющий слой (Global Average Pooling).

Выходной слой (Output Layer):

Полносвязный слой (Dense) с 2 нейронами, соответствующими двум классам (незрелая и зрелая катаракта), с функцией активации softmax для получения вероятностей классов.

Параметры модели оптимизируются методом SGD с начальной скоростью обучения 0.1 и функцией потерь кросс-энтропия (categorical cross-entropy), что позволяет минимизировать разницу между предсказанными и истинными значениями классов. Для адаптивного уменьшения скорости обучения использовался метод lr_scheduler.StepLR с шагом уменьшения каждые 10 эпох и коэффициентом уменьшения 0.1.

B. VGG16

Архитектура модели

Модель для классификации катаракты основана на архитектуре VGG16 с двумя дополнительными сверточными слоями [11]. Входной слой предполагается размером, соответствующим размеру изображений в наборе данных. Основная часть модели состоит из сверточных слоев, слоев пулинга и полносвязных слоев, а также двух дополнительных сверточных слоев, которые добавлены после архитектуры VGG16. Схема архитектуры модели представлена на рисунке 6.

Входной слой (Input Layer):

Входные изображения размером 96×96×3

Сверточные слои VGG16:

VGG16 состоит из сверточных слоев, включая слои с активацией ReLU и слои пулинга.

Каждый сверточный слой обычно следует за активацией ReLU, за которой следует слой пулинга. Это помогает извлекать признаки из изображений на разных уровнях абстракции.

1. Дополнительные сверточные слои: после базовой архитектуры VGG16 добавлены два дополнительных сверточных слоя. Эти слои могут быть специально настроены для более глубокого изучения признаков, уточнения представлений и улучшения производительности модели.
2. Активации ReLU: после каждого сверточного слоя (включая дополнительные слои) идут активации ReLU, которые выполняют операцию поэлементного увеличения входных данных. Это помогает вводить нелинейность в модель и делает ее более способной к обучению сложных функций [12].
3. Пулинговые слои: после некоторых сверточных слоев следуют слои пулинга, которые уменьшают размер признаковых карт и помогают модели стать более инвариантной к масштабированию исходных данных.

Добавление двух дополнительных сверточных слоев после архитектуры VGG16 может улучшить способность модели изучать сложные признаки из изображений и повысить ее точность в задачах классификации или других задачах компьютерного зрения.

Выходной слой (Output Layer):

Полносвязный слой (Dense) с 2 нейронами, соответствующими двум классам (незрелая и зрелая катаракта), с функцией активации softmax для получения вероятностей классов.

Параметры модели оптимизируются методом SGD и функцией потерь кросс-энтропия (categorical cross-entropy), а также используется lr_scheduler.StepLR для адаптивного уменьшения скорости обучения.

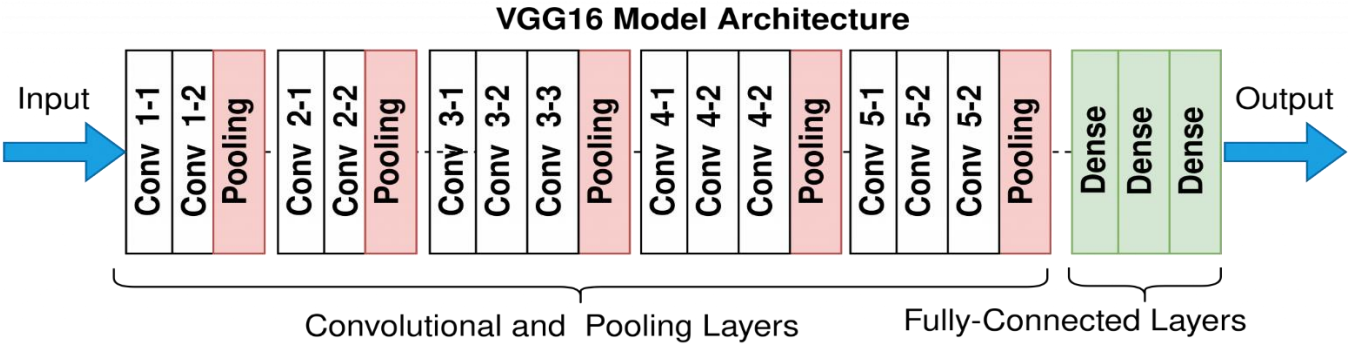


Рис. 6. Архитектура VGG16

III. Результаты

Для оценки точности моделей была использована метрика Accurate [13, 14], которая представляет собой отношение количества правильных предсказаний на общее количество предсказаний.

$$\boldsymbol{Accurate} = \frac{\text{Количество правильных предсказаний}}{\text{Общее количесво предсказаний}}$$

Более формально, если y_i – истинные метки классов, а $\hat{y}i$ - предсказанные метки классов, и n – общее количество наблюдений, то формула для вычисления точности выглядит следующим образом:

$$Accurate = \frac{1}{n}\sum_{i=1}^{n} 1\,(y_i = \hat{y}i),$$

где $1(y_i = \hat{y}i)$ – индикаторная функция, которая равна 1, если предсказанное значение $\hat{y}i$ совпадает с истинным значением y_i, и 0 в противном случае.

A. *Resnet50*

1. Процесс обучения.

Модель обучалась на тренировочных данных в течение 20 эпох. Время обучения составило примерно 45 минут. График обучения модели показан на рисунке 7.

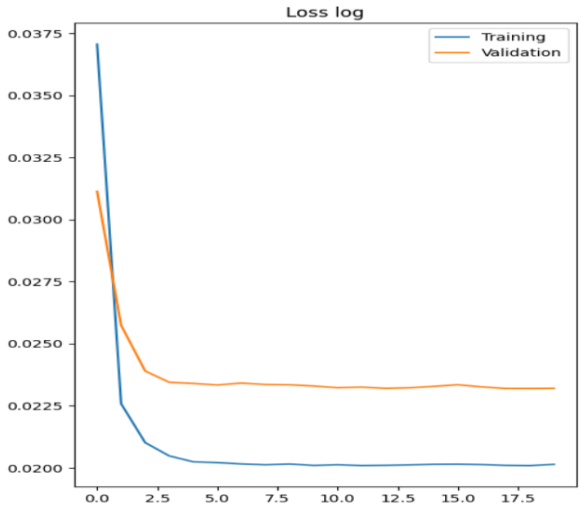


Рис. 7. График обучения Resnet50

2. Результаты обучения

По результатам обучения на протяжении 20 эпох модель продемонстрировала следующие показатели:

1. Loss на тренировочной выборке: 0.0201;
2. Accuracy на тренировочной выборке: 0,98;
3. Loss на валидационной выборке: 0.0232;
4. Accuracy на валидационной выборке: 0,97.

Матрица ошибок отображена на рисунке 8.

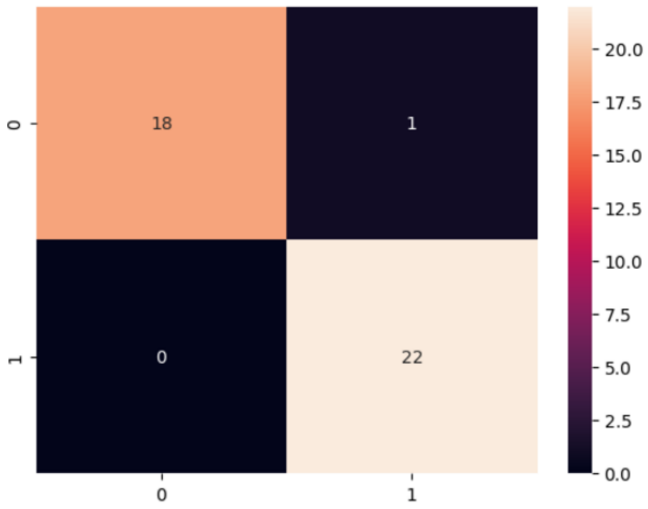


Рис. 8. Матрица ошибок Resnet50

B. *VGG16*

1. Процесс обучения.

Модель обучалась на тренировочных данных в течение 20 эпох. Время обучения составило примерно 1,5 часа. График обучения модели показан на рисунке 9.

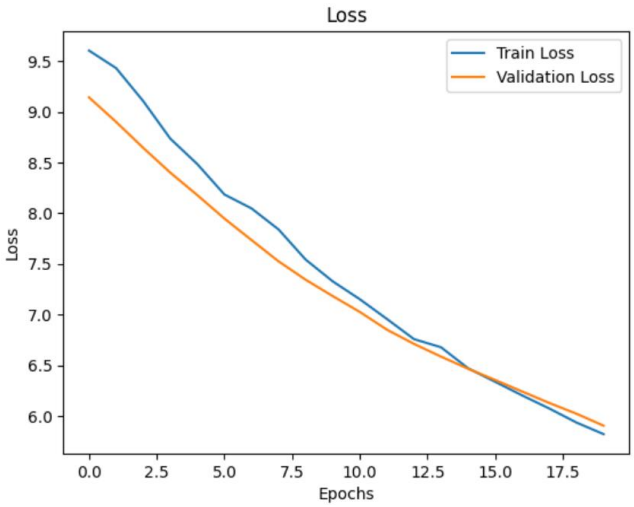


Рис. 9. График обучения VGG16

2. Результаты обучения

По результатам обучения на протяжении 20 эпох модель продемонстрировала следующие показатели:

1. Loss на тренировочной выборке: 0.1303;
2. Accuracy на тренировочной выборке: 0.90;
3. Loss на валидационной выборке: 0.1738;
4. Accuracy на валидационной выборке: 0.86.

Матрица ошибок отображена на рисунке 10.

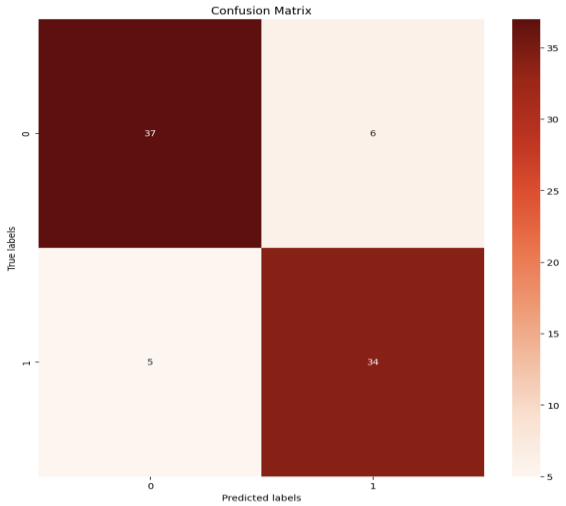


Рис. 10. Матрица ошибок Resnet50

Accuracy на валидационной выборке для каждой модели отображена в таблице 1.

Таблица 1. Accuracy каждой модели

Модель	Accuracy
Resnet50	0,97
VGG16	0.86

Из анализа таблицы видно, что лучшие результаты достигла модель Resnet50, демонстрируя точность в значении 0.97. Этот показатель указывает на высокую точность классификации модели.

Модель VGG16 также продемонстрировала хорошие результаты, показав точность 0.86. В то время как точность модели VGG16 ниже, чем у Resnet50, она все равно представляет собой значимое достижение в задаче классификации катаракты.

Обе модели успешно распознают и классифицируют катаракты глаза. Визуализация результатов предсказаний моделей подтверждает их способность точно определять наличие катаракты на изображениях глаз.

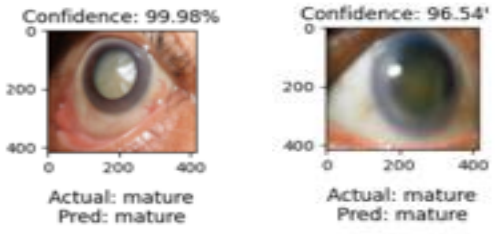


Рис. 11. Предсказанные классы катаракты при помощи ResNet и VGG16.

Как видно из рисунка 11, обе модели успешно классифицируют катаракту глаза. Визуализация демонстрирует высокую точность модели, что подтверждается низким значением среднеквадратичного отклонения (RMSE).

IV. Заключение

В рамках данного исследования был проведен анализ и сравнительное изучение двух нейронных сетей, применяемых для классификации катаракты на медицинских изображениях глаз.

Были изучены архитектура каждой модели, чтобы понять ее структуру и принцип работы; процесс обучения, чтобы узнать, как модели обучались и оптимизировались для достижения максимальной точности. Была проведена оценка качества работы моделей и сравнены результаты.

Анализ показал, что модель ResNet50 демонстрирует более высокую точность классификации катаракты по сравнению с моделью VGG16. Это свидетельствует о преимуществах использования современных архитектур нейронных сетей, таких как ResNet50, в области медицинской диагностики и распознавания патологий глаз.

Полученные результаты являются важным вкладом в разработку систем компьютерного зрения для диагностики катаракты и других заболеваний глаз.

Литература

[1] Hammond C. J. et al. The heritability of age-related cortical cataract: the twin eye study //Investigative ophthalmology & visual science. – 2001. – Т. 42. – №. 3. – С. 601-605.

[2] Sramka M. et al. Improving clinical refractive results of cataract surgery by machine learning //PeerJ. – 2019. – Т. 7. – С. e7202.

[3] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[4] 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.

[5] 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.

[6] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95.10.21146/0042-8744-2022-3-93-105.

[7] Cataract Classification Dataset, available at: https://www.kaggle.com/datasets/akshayramakrishnan28/cataract-classification-dataset
(Accessed: October 05, 2024).

[8] Çınar A., Yıldırım M., Eroğlu Y. Classification of pneumonia cell images using improved ResNet50 model //Traitement du Signal. – 2021. – Т. 38. – №. 1. – С. 165-173.

[9] Albashish D. et al. Deep CNN model based on VGG16 for breast cancer classification //2021 International conference on information technology (ICIT). – IEEE, 2021. – С. 805-810.

[10] Lin D. et al. A practical model for the identification of congenital cataracts using machine learning //EBioMedicine. – 2020. – Т. 51.

[11] Zhang X. Q. et al. Machine learning for cataract classification/grading on ophthalmic imaging modalities: A survey //Machine Intelligence Research. – 2022. – Т. 19. – №. 3. – С. 184-208.

[12] Pietikäinen M. et al. Computer vision using local binary patterns. – Springer Science & Business Media, 2011. – Т. 40.

[13] Vujović Ž. et al. Classification model evaluation metrics //International Journal of Advanced Computer Science and Applications. – 2021. – Т. 12. – №. 6. – С. 599-606.

[14] Liu Y. et al. Toward a better understanding of model validation metrics. – 2011.

ГРНТИ 28.23.15
ASJC 1702
OECD 01.02.EP

Генерация оптического потока с помощью машинного обучения

Д.А. Подгорный
Кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m2314956@edu.misis.ru

Селезенёв, Иван
Кафедра инженерной кибернетики НИТУ «МИСиС»
Москва, Россия
m1902948@edu.misis.ru

***Аннотация* — данная работа посвящена обзору развития решений задачи нахождения оптического потока методами машинного обучения. В её рамках рассматриваются ключевые методы оценки оптического потока, изучается их эволюция с течением времени и сравнивается эффективность их работы.**

Ключевые слова — оптический поток, видеопоследовательность, поле корреляции, рекуррентные нейронные сети, трансформеры, RAFT, GMFlowNet.

I. Введение

Под задачей нахождения оптического потока на последовательности изображений понимается нахождение смещения пикселей, соответствующих подвижным элементам, изменяющим свою позицию с течением времени. Этот метод крайне полезен для анализа сцены, о которой отсутствует какая-либо информация за исключением непосредственно видеоряда. Используя данные оптического потока изображения, можно осуществлять трекинг элементов, моделировать траекторию движения объектов внутри сцены, вычислять промежуточные кадры, искусственно увеличивая кадровую частоту видео.

Задача вычисления оптического потока исследуется с 80-х годов прошлого века. За это время были разработаны различные подходы к его вычислению: изначально оптический поток вычислялся решением задачи оптимизации, внутри которой рассчитывались схожие участки изображения и определялись параметры их видоизменения. Такой подход снискал успех, однако его дальнейшее развитие было затруднительно из-за сложности математических расчётов, необходимых для обеспечения универсальности и надёжности в прикладных условиях.

С развитием глубокого обучения стали изучаться альтернативные подходы к решению задачи нахождения оптического потока [1].

Классические ИИ-модели для нахождения оптического потока между двумя изображениями работают за счёт обнаружения шаблонных признаков и построения на их основе поля корреляции – для каждого пикселя первого изображения строится поле корреляции со всеми пикселями на втором изображении (рис. 1) [2].

В результате данной операции образуется 4-мерный тензор, значения которого передаются в последующие слои нейросети, которая по полученным признакам должна посчитать карту смещений для каждого пикселя на изображении.

Такой подход зарекомендовал себя как более эффективный чем традиционные методы, и поэтому множество исследований ориентировано на его развитие [3].

Ключевым вопросом для дальнейших исследований является разработка эффективных архитектур, которые лучше работают, легче поддаются обучению и хорошо адаптируются к новым сценам.

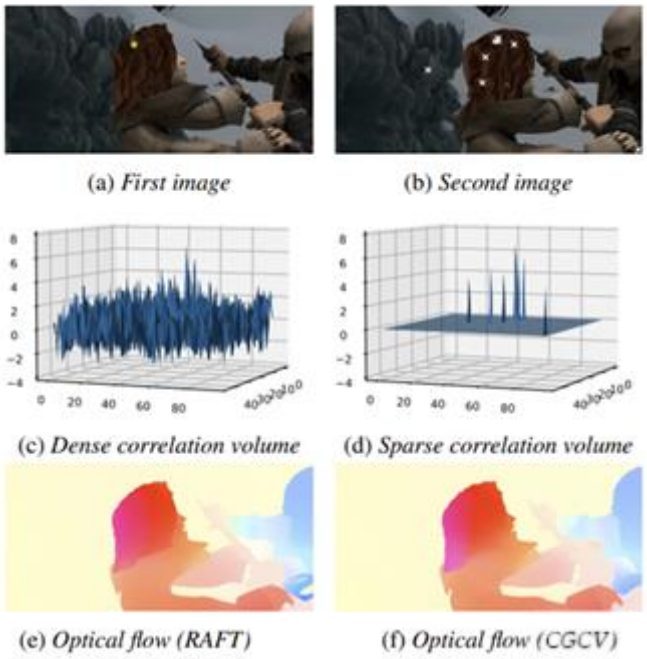


Рис. 1. Построение различных полей корреляции для нахождения оптического потока e и f между изображениями a и b, где c и d - поля корреляции

II. Обзор RAFT

A. Описание

В RAFT [4] основная концепция ИИ-модели для вычисления оптического не поменялась, но изменения коснулись архитектуры: её структуры, слоёв и нюансов работы.

Была разработана модификация модели GRU – ConvGRU, в которой все полносвязные слои заменены свёрточными. Помимо этого, на практике RAFT

обрабатывает изображения высокого разрешения, в то время как SOTA-решения того времени использовали сжатую версию изображения для распознавания оптического потока, который затем преобразовывался в тензор большего разрешения, в результате чего терялись данные о перемещении менее явных элементов изображения [5].

За счёт внедрённых изменений, авторам удалось предложить модель распознавания оптического потока, которая лучше работает, легче поддается обучению и хорошо адаптируется к новым сценам. RAFT обладает следующими преимуществами перед предшественниками:

- Лучшая точность: метрики демонстрируют уменьшение ошибки F1-all на 16% и EPE на 30% в сравнении с предыдущими рекордсменами.

- Универсальность: RAFT, обученный исключительно на синтетических данных, достигает EPE равной 5.04 пикселя на датасете KITTI, изображения для которого получены естественным образом. Это на 40% меньше, чем предыдущая лидирующая модель, обученная на тех же данных.

- Улучшенная обучаемость: для обучения модели требуется в 10 раз меньше итераций, чем предшественникам.

- Скорость работы: RAFT обрабатывает видео в разрешении 1088x436 пикселей со скоростью 10 кадров в секунду на графическом процессоре 1080Ti. Облегчённый RAFT с в пятеро раз меньшим количеством параметров достигает частоты обработки в 20 кадров в секунду.

На рисунке 2 представлена архитектура RAFT. RAFT состоит из 3 основных компонентов:

3) GRU-блока, рекуррентно обновляющего оптический поток, используя текущие оценку для поиска значений из набора корреляционных тензоров. Дизайн RAFT черпает вдохновение из множества существующих работ, но в значительной степени новаторский.

Во-первых, RAFT отказывается от подхода "coarse-to-fine"[6], где сначала поток оценивается в грубых значениях, а затем значения уточняются с большей детальностью. Вместо этого используются методы с получением выпуклой комбинации и взвешиванием маски, полученной свёрткой (см. далее). Такое изменение позволяет снизить ошибку и повысить способность модели корректно обрабатывать наиболее малые объекты сцены.

Во-вторых, RAFT – одна из первых рекуррентных моделей для определения оптического потока. Авторы приводят в пример IRR[7] как единственный известный им на тот момент рекуррентный подход для такого рода задач, и указывают на недостатки предложенных в нём решений. Так, механизм памяти в предложенных сетях ограничен либо пятью итерациями, либо количеством уровней пирамиды, в зависимости от того, используется FlowNetS[8] или PWC-Net[9]. В противовес, предложенная модель имеет значительно меньше параметров, и в то же время механизм памяти работает исправно при более 100 итераций.

В-третьих, в блоке обновления используется модифицированный GRU-слой, способный работать с четырёхмерными тензорами, в отличие от предшественников, способных работать с лишь с простыми свёрточными / корреляционными сломи.

Ранее предлагалось рассматривать оптический поток как задачу дискретной оптимизации с использованием глобальной целевой функции (using a global objective).

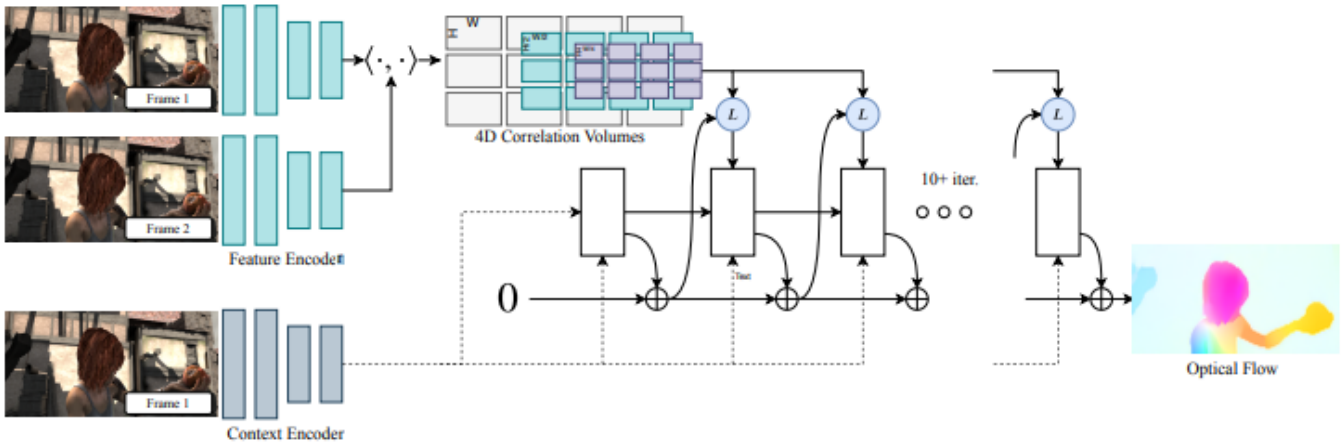


Рис. 2. Архитектура RAFT

1) Кодировщика признаков, который извлекает признаки для каждого пикселя из обоих входных изображений, и контекстный кодировщик, который извлекает признаки только из первого изображения I_1.

2) Корреляционного слоя, который вычисляет поле корреляции W × H для каждого пикселя, образуя корреляционный 4D-тензор W × H × W × H. Из полей корреляции затем строится пирамида корреляции в различных масштабах: 1x, 1/2x, 1/4x, 1/8x.

Одной из проблем этого подхода является огромный размер пространства поиска, так как каждый пиксель может быть напрямую сопоставлен с тысячами пикселей на другом кадре. Менц и др. смогли сократить пространство поиска, используя описатели признаков [10].

DCFlow [11] представил другие улучшения – например, использование нейронной сети для описания признаков, и построение 4D-тензора всех пар признаков.

Этот тензор затем обрабатывался с помощью алгоритма полуглобального согласования (SGM).

RAFT, как и DCFlow также строит 4D-тензор по полученным признакам. Однако, вместо обработки объемов затрат с помощью алгоритма SGM, RAFT использует нейронную сеть для оценки потока. Использование нейросети для обработки тензора означает возможность обучать её вместе с остальной моделью, напрямую влияя на ход, и, как следствие, качество обучения.

Многие модели используют обучение методом градиентного спуска. RAFT избегает этого подхода с целью ускорения вычислений. Вместо этого градиентный спуск имитируется большим количеством блоков обновлений, которые извлекают характеристики из корреляционных тензоров, чтобы определить спуск.

B. Методы

Для пары изображений I_1, I_2 оценивается поле смещений $(f1, f2)$, которое сопоставляет каждый пиксель с координатами (u, v) на изображении I_1 соответствующим координатам (u', v')на изображении I_2, что представлено в формуле (1).

$$(u', v') = \big(u + f_1(u), v + f_2(v)\big). \quad (1)$$

Метод RAFT, как было сказано ранее, можно разбить на три этапа:

1) *Получение характеристик.*
2) *Построение тензора корреляций.*
3) *Блок обновлений.*

Формула (2) иллюстрирует получение характеристик, которое выполняется следующим образом: кодировщик g_Θ отображает RGB-изображение в пространство характеристик.

$$R^{H \times W \times 3} \rightarrow R^{H/8 \times W/8 \times D}, \quad (2)$$

где $D = 256$.

Тензор корреляции C строится по признакам из кодировщика $g_\Theta(I_1)$ и $g_\Theta(I_2)$ путём их скалярного произведения. На основе этого тензора строится пирамида корреляции $\{C1, C2, C3, C4\}$, получаемая свёрткой C фильтрами размеров 1, 2, 4 и 8 соответственно.

На каждом уровне пирамиды корреляции для каждого пикселя x на I_1 ищется наиболее коррелирующий пиксель x' на I_2 в заданном радиусе. Радиус фиксирован, поэтому на свёрнутых слоях пирамиды C^k поиск коррелирующего пикселя будет охватывать большую площадь с увеличением k. Таким образом, поле корреляции определяется на микро- и макроуровнях, а затем конкатенируется в единое пространство характеристик.

Блок обновления работает следующим образом. Определяется последовательность оценок потока $\{f_1, \dots, f_N\}$ для каждого пикселя с начальной точкой $f_0 = 0$. Каждая итерация начинается с получения оценки потока, поля корреляции и скрытого состояния нейрона модифицированного GRU-слоя, в котором полностью подключенные слои заменены на свертки.

Веса слоя связаны между итерациями, а слой натренирован таким образом, чтобы обеспечить сходимость к фиксированной точке

Процесс одного обновления может быть разделен на следующие действия: оператор обновления принимает текущую оценку потока f_k корреляционные признаки и скрытое состояние, на основе этих входных данных оператор вычисляет корректировку Δf, которая применяется к текущей оценке потока $f_{k+1} = \Delta f + f_{k+1}$. Скрытое состояние, полученное из GRU, передается через два сверточных слоя для предсказания обновления потока Δf.

Выходной поток имеет разрешение 1/8 от исходного изображения, которое затем повышается до полного путём получения для каждого пикселя выпуклой комбинации соседних пикселей внутри сетки размером 3x3. Используется два свёрточных слоя для получения маски H/8 x W/8 x 9 и вычисляем softmax от 9 соседей.

Конечное поле потока с высоким разрешением определяется с помощью маски для получения взвешенной комбинации по окрестности, затем перестановки и изменения формы в поле потока размером $H \times W \times 2$.

При заданном истинном потоке f_{gt} последовательности оценок потока $\{f_1, \dots, f_N\}$, функция потерь представлена в формуле (3).

$$L = \sum_{i=1}^{N} \gamma^{N-i} \left| \left| f_{gt} - f_i \right| \right|_1, \quad (3)$$

где γ – изменяемый параметр, оптимальное значение которого 0,8.

III. Обзор GMFlowNet

A. Описание

GMFlowNet [12] была разработана с целью разрешить проблему низкой точности предшествующих моделей (таких, как RAFT) в сценариях, когда оптический поток вычисляется между кадрами со значительным смещением.

Для решения этой проблемы авторы GMFlowNet внесли следующие новшества в современную архитектуру модели для определения оптического потока:

1) Изменение кодировщика модели с применением архитектуры трансформера с модифицированным блоком внимания.

2) Совместное использование полей корреляции и полей сопоставлений, вычисленных на их основе, для предсказания потока.

С момента появления RAFT, вышло множество работ об эффективности архитектуры трансформеров в задаче нахождения оптического потока. Таким образом, вместо классического кодировщика характеристик изображения в GMFlowNet применяется кодировщик-трансформер. Авторы вносят исправления во взятую за основу архитектуру, а именно – в блок внимания, чтобы снизить потерю информации в процессе обработки данных.

Также авторы нашли полезным вычислять поля сопоставлений на основе полей корреляции и

использовать полученные данные в комбинации для достижения большей точности модели.

Таким образом удалось увеличить точность модели и устранить недостатки предыдущих архитектур-трансформеров.

B. *Методы*

Модель состоит из трёх основных частей: кодировщика характеристик (Large Context Feature Extraction), блока сопоставлений (Global Matching) и блока оптимизации (Optimization Operator) – см. рис. 3.

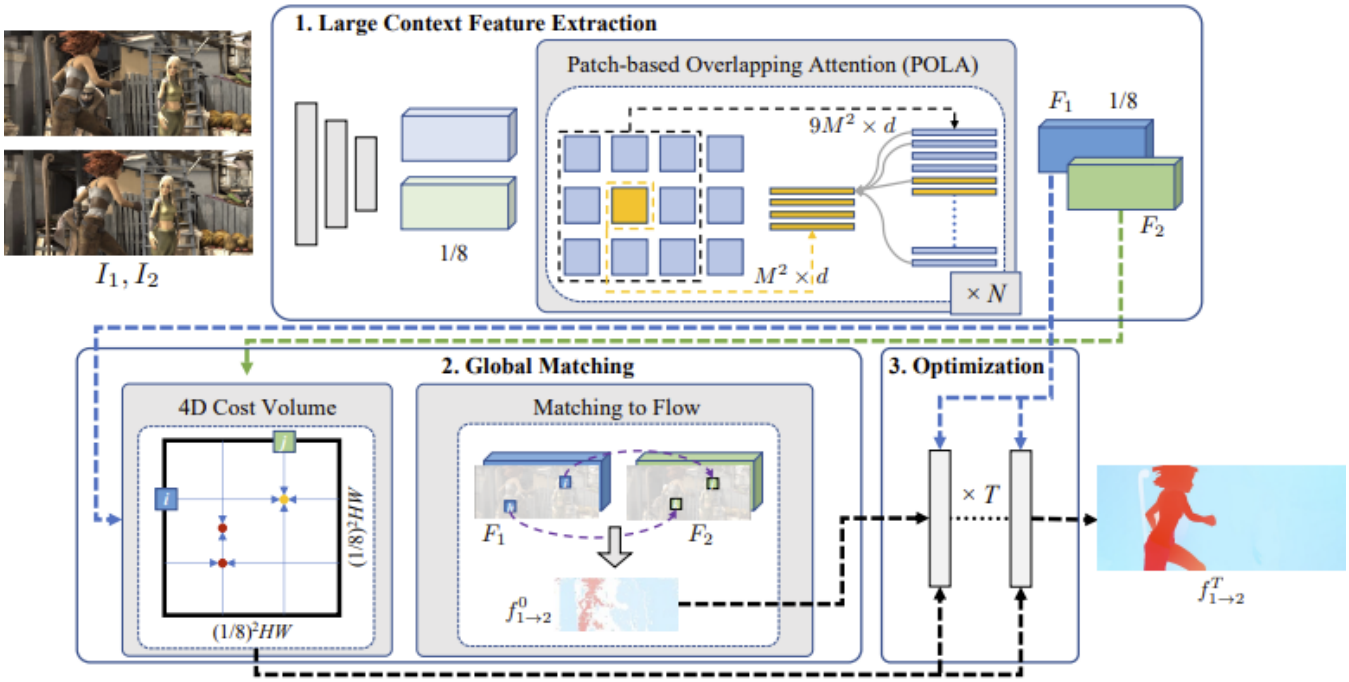


Рис. 3. Архитектура GMFlowNet

Large Context Feature Extraction. Обработка контекста – это ключевой фактор в сопоставлении неоднозначных участков изображения, таких как репетитивные узоры и регионы без отсутствующей текстуры. Для определения контекста используется последовательность из трёхслойной свёрточной сети для выделения характеристик и сети-трансформера, обрабатывающей их.

Из-за вычислительной сложности классического блока внимания внутри трансформера его крайне невыгодно применять ко всему пространству полученных характеристик. Чтобы снизить стоимость исчисления, авторы разработали облегчённый блок внимания под названием patch-based overlapping attention – POLA.

POLA разбивает характеристики изображения на непересекающиеся фрагменты размером MxMxd, где M – размерность фрагмента в пикселях, а d – количество характеристик, и для каждого фрагмента применяет механизм внимания на него и на его соседей, получая в результате карту внимания размера MxMxd.

Авторы обращают внимание, что предшествующие работы, например [13], использовали механизм «смещения окна», который требовал два отдельных блока внимания, что вело к потере контекстуальной информации. В GMFlowNet используется один блок внимания, что улучшает точность, занимает меньше памяти и улучшает производительность.

Global Matching. Из извлечённых из двух изображений характеристик F_1 и F_2 строится корреляционный тензор C по формуле (4):

$$C(i,j,u,v) = F_1(i,j)\dot{F}_2(u,v), \quad (4)$$

где (i,j) и (u,v) – это позиция в F_1 и F_2 соответственно.

Для каждой позиции в F_1 вычисляется близость с позициями из F_2 как представлено в формуле (5).

$$P_c(i,j,u,v) = \mathrm{softmax}\big(C(i,j,\cdot)\big) \odot \mathrm{softmax}\big(C(\cdot,u,v)\big), \quad (5)$$

где $\odot$ – попиксельное произведение.

Далее считаются все совпадения позиций из изображения 1 в изображении 2 как, формула (6).

$$\mathrm{M}_{1\to2}(i,j) = arg\,max_{u,v}\, P_c(i,j,u,v). \quad (6)$$

Аналогичным образом считаются все совпадения из второго изображения в первое $M_{2\to1}$.

Набор совпадающих пикселей M_c можно воспринимать как конъюнкцию множеств $M_{1\to2}$ и $M_{2\to1}$, формула (7)

$$M_C = \{(\hat{i},\hat{j}) \big| (\hat{i},\hat{j}) = M_{2\to1}\big(M_{1\to2}(\hat{i},\hat{j})\big)\}, \quad (7)$$

А изначальный оптический поток вычисляется как представлено в формуле (8).

$$f^0_{1\to2} = \begin{cases} M_{1\to2}(i,j) - (i,j), (i,j) \in M_C \\ (0,0), \text{ в противном случае.} \end{cases} \quad (8)$$

Optimization Operator. Авторы используют готовый оператор из сети RAFT, за тем исключением, что первая оценка потока $f^0_{1\to2}$ инициализируется не нулевыми значениями, а результатом поиска совпадений global matching.

IV. Метрики и сравнение

Для анализа модели было принято решение создать синтетический датасет с помощью средств для трёхмерного моделирования. Blender обладает инструментарием для экспорта данных о движении в сцене в виде изображений формата OpenEXR – см. рис. 4.

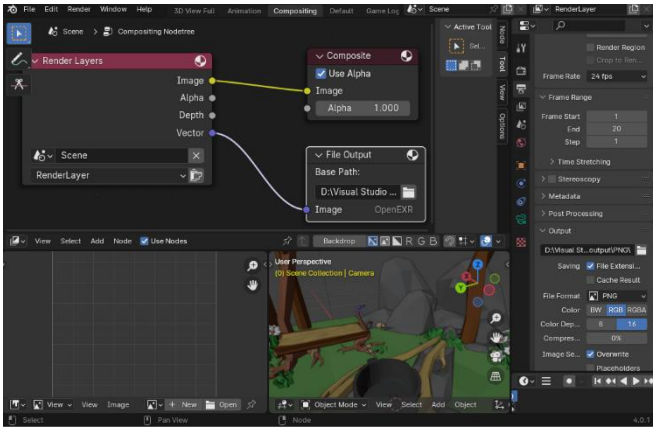


Рис. 4. Экспорт оптического потока из Blender

Для обработки OpenEXR-изображений используется python с одноименным модулем. Подробную информацию о том, как генерировать синтетические данные для тестов оптического потока, можно найти в указанной статье [14].

A. *Расчёт метрик*

Для обработки OpenEXR-изображений используется python с одноименным модулем.

В данном представлении красный и зелёный цветовые каналы являются смещениями пикселей по горизонтали и вертикали соответственно. При этом для видеопоследовательности $I(I_1, I_2, \dots I_n)$ вычисляется набор OpenEXR-изображений $I_B(I_{B1}, I_{B2}, \dots I_{Bn})$ таких, что оптический поток между двумя изображениями записывается как представлено на формуле (9).

$$f(I_i, I_{i+1}) = I_{B\,i+1}. \quad (9)$$

Следовательно, I_{B1} не несёт в себе какой-либо полезной информации. Формула (10) представляет вычисления RAFT оптического потока для пары изображений.

$$(I_i, I_{i+1}) = I_{Ri}. \quad (10)$$

Таким образом, результатом работы RAFT является множество тензоров оптического потока, обладающее размером, вычисляемым по формуле 11.

$$|I_R| = n - 1. \quad (11)$$

Соответственно, отклонение между множествами I_B (ground truth) и I_R (prediction) по осям x и y можно рассчитать, по формулам (12) и (13) соответственно.

$$\Delta I_{i_x} = I_{Bi_x} - I_{Ri_x} \quad (12)$$

$$\Delta I_{i_y} = I_{Bi_y} - I_{Ri_y} \quad (13)$$

Тогда ошибка прогноза EPE (end-point-error) для пары изображений (I_i, I_{i+1}) с количеством пикселей M вычисляется как представлено в формуле (14).

$$EPE_i = \frac{1}{M}\sum_{k=1}^{M}\sqrt{(\Delta I_{xik})^2 + (\Delta I_{yik})^2}. \quad (14)$$

И EPE для видеопоследовательности определяется как среднее арифметическое, формула (15).

$$EPE = \sum_{i=1}^{N-1} EPE_i \quad (15)$$

Метрика EPE исчисляется в пикселях, т.е. она зависима от размера изображения. Авторы RAFT записывали метрики на наборе данных KITTI [15], изображения внутри которого обладают разрешением 1280 на 344 пикселя. Чтобы сопоставить метрики, полученные собственным путём с заявленными, тестовые изображения из множества I_B записываются в таком же разрешении – см. рис. 5.

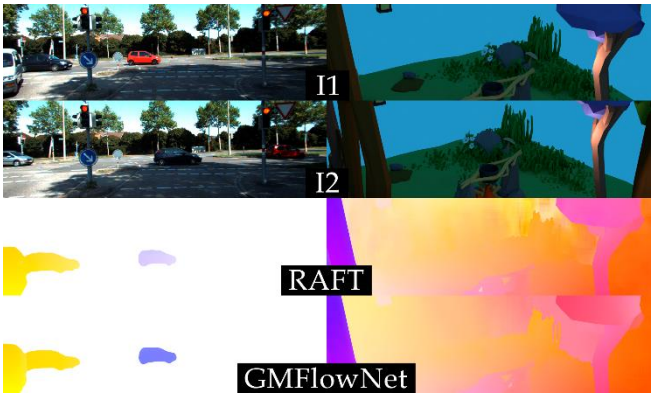


Рис. 5. Демонстрация работы двух моделей - RAFT и GMFlowNet на паре изображений из набора KITTI (слева) и из самостоятельно синтезированных данных (справа)

При тесте на 200 собственных изображениях показатель EPE для RAFT составил 15.45 пикселя, в то время как на изображениях KITTI – 5.01, а на изображениях Sintel [16] – 5.67.

Для GMFlowNet EPE составил 15.28 на собственном наборе данных, 4.24 на KITTI, 4.78 на Sintel.

Бенчмарк проводился на тренированных моделях

B. *Сравнение результатов*

Такое сильное расхождение в метриках на собственном наборе данных при едином разрешении связано с тем, что в рамках эксперимента было решено синтезировать кадры с намеренно высоким смещением, чтобы пронаблюдать разницу между RAFT и GMFlowNet и подтвердить лучшую адаптивность последней модели к таким смещениям.

Но пусть фактически результат бенчмарка более новой модели действительно лучше, разница между показателями находится на уровне погрешности. Тем не менее, в тестах KITTI и Sintel, архитектура трансформера однозначно демонстрирует своё превосходство.

V. Заключение

Исследования в области определения оптического потока методами машинного обучения активно ведутся: пробуются различные архитектуры и разыскиваются пути оптимизации существующих решений. Последние изыскания на эту тему освещают преимущества применения сетей-трансформеров в данной прикладной

задаче и задают вопросы, на которые ещё предстоит ответить.

Современные методы совершенствуют устройство кодировщика и процессы построения полей корреляции, оставляя процесс оптимизации потока неизменным с момента публикации RAFT. Вполне вероятно, что, исчерпав возможные пути совершенствования перечисленных компонентов, типичных для большинства современных сетей для распознавания оптического потока, исследователи перейдут к изучению более скоростных методов оптимизации потока, либо вовсе сменят подход в связи с развитием технологий машинного обучения.

Список литературы

[1] R. Sadekov, A. Popov, M. Zolotov, R. Bikmaev, “Improving the Accuracy of Supporting Mobile Objects with the Use of the Algorithm of Complex Processing of Signals with a Monocular Camera and LiDAR,” in 26th Saint Petersburg International Conference on Integrated Navigation Systems, 2019, pp. 1-4.

[2] D. B. Pazychev and R. N. Sadekov, "Simulation of INS Errorsof Various Accuracy Classes," 2020 27th Saint PetersburgInternational Conference on Integrated Navigation Systems(ICINS), St. Petersburg, Russia, 2020, pp. 1-3

[3] Sadekov R. N. et al. “Road sign detection and recognition inpanoramic images to generate navigational maps,” 24th Saint Petersburg International Conference on IntegratedNavigation Systems (ICINS). – IEEE, 2017, pp. 1-5

[4] Zachary Teed and Jia Deng. “RAFT: recurrent all-pairs field transforms for optical flow”, CoRR abs/2003.12039, 2020.

[5] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-CostNavigation System for UAV," 30th Saint PetersburgInternational Conference on Integrated Navigation Systems(ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6.

[6] A. Jahedi, M. Luz, M. Rivinius, A. Bruhn, “Ccmr, High resolution optical flow estimation via coarse-to-fine context-guided motion reasoning,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6899–6908, 2024.

[7] J. Hur, S. Roth, “Iterative residual refinement for joint optical flow and occlusion estimation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5754–5763, 2019.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, ‘‘FlowNet: Learning optical flow with convolutional networks,’’ in proceedings of IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2758–2766.

[9] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, ‘‘PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,’’ in proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8934–8943.

[10] M. Menze, C. Heipke, A. Geiger, “Discrete optimization for optical flow” in German Conference on Pattern Recognition, pp. 16–28, 2015.

[11] J. Xu, R. Ranftl, and V. Koltun, “Accurate optical flow via direct cost volume processing,” in proceedings of IEEE Conferenece on Computer Vision and Pattern Recognition (CVPR), ArXiv Preprint ArXiv:1704.07325. 2017, 2017.

[12] S. Zhao, L. Zhao, Z. Zhang, E. Zhou, and D. Metaxas, ‘‘Global matching with overlapping attention for optical flow estimation,’’ in proceedings of IEEE/CVF Conferenece on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 17571–17580.

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, “Swin Transformer: Hierarchical vision transformer using shifted windows,” in International Conference on Computer Vision (ICCV), 2021, pp. 10012–10019.

[14] “Groundtruth data for Computer Vision with Blender”, available at: https://www.tobias-weis.de/groundtruth-data-for-computer-vision-with-blender (Accessed: April 27, 2024).

[15] “KITTI Vision Benchmark Suite”, available at: https://www.cvlibs.net/datasets/kitti/index.php (Accessed: April 30, 2024).

[16] “MPI Sintel Flow Dataset”, available at: http://sintel.is.tue.mpg.de (Accessed: May 14, 2024).

ГРНТИ: 28.23.15
ASJC: 1707
OECD: 01.02.EP

Распознавание поз нескольких объектов на изображении с использованием real-time моделей

Д. А. Личко
кафедра инженерной кибернетики
НИТУ МИСИС
Москва, Россия
m1902984@edu.misis.ru

Д. А. Рамзайцев
кафедра инженерной кибернетики
НИТУ МИСИС
Москва, Россия
m1904484@edu.misis.ru

***Аннотация* — Данная работа посвящена исследованию эффективности применения глубоких нейронных сетей для решения задачи определения поз нескольких объектов (людей, животных) на изображении. В рамках работы были проанализированы основные подходы и методики к решению задачи. В частности были обучены и проанализированы real-time модели RMTPose и HRNet. В работе представлено сравнение эффективности моделей на различных изображениях. В качестве данных использовались датасеты COCO tiny, COCO + ubody с изображениями людей, а также датасет AP-10K с изображениями животных.**

Ключевые слова — компьютерное зрение, CV, распознавание позы, сверточные нейронные сети, глубокое обучение, CNN

I. Введение

В настоящее время нейронные сети становятся все более востребованным инструментом в различных областях человеческой деятельности. Искусственный интеллект, способный обучаться на данных и принимать решения на основе опыта, проникает в самые разнообразные сферы нашей жизни.

Искусственные нейронные сети широко распространены в разных отраслях: в обнаружение машин и предсказании их передвижения [1], в оценке поз в робототехнике [2], управлении умным городом [3], навигации [4] и распознавании текста [5].

Одним из ярких примеров успешного применения нейронных сетей является область компьютерного зрения.

Одна из задач, которая эффективно решается с помощью моделей машинного обучения - это задача определения позы человека на изображении (human pose estimation). Это задача подразумевает нахождение координат и ориентации ключевых точек на человеческом теле в двумерном пространстве. Ключевые точки могут описывать такие элементы как шея, плечи, локти, запястья, таз, колени, голени и др. Определение точного положения этих ключевых точек позволяет компьютеру понимать позу человека на изображении, его движения и жесты.

Оценка позы человека имеет широкий спектр применений, начиная от автоматического анализа движений в спорте, например, для анализа техники упражнений, заканчивая медициной, виртуальной реальностью и системами наблюдения в общественных местах.

В данной статье будет рассмотрено применение нейронных сетей для задачи оценки позы нескольких объектов на изображении, их особенности, преимущества и недостатки. В частности будут обучены и оценены SOTA модели HRNet и RTMPose.

II. Анализ источников

Задача определения позы объекта - это способ определения координат ключевых точек, например, руки, головы и т.д., а также связей между ними (пары). Таким образом задача определения позы сводится к построению скелетоподобного описания тела человека. Пример определения позы представлен на рисунке 1.

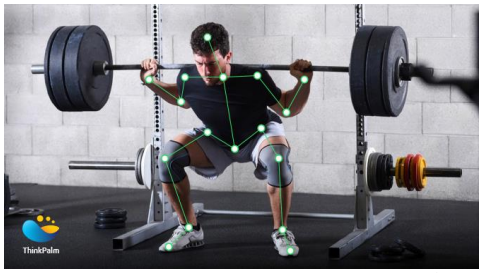


Рис. 1. Пример определения позы человека на изображении

Существует 3 методики моделирования позы объекта (их визуализация представлена на рисунке 2):

1. Модель на основе скелета;
2. Модель на основе контура;
3. Модель на основе объёма.

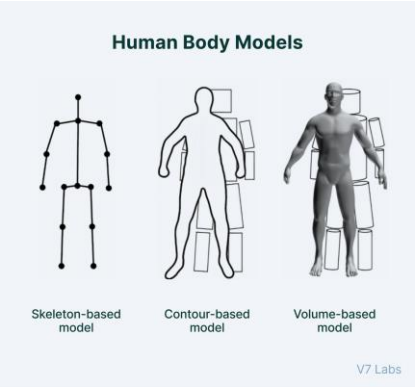


Рис 2. Методики моделирования позы

В данной работе будут рассмотрены методы, позволяющие создавать модели на основе скелета.

Основные методы решения данной задачи можно разделить на 3 основные группы:

1. Решения с использованием маркеров. На объекте определения позы устанавливаются контрастные метки, например, белые шарики. Далее на изображении/видео эти маркеры соединяются по особым правилам, чтобы получился скелет. Этот метод не требует сложных технических решений, а также довольно точен. Однако он требует подготовки и установки маркеров. Данный метод широко используется при производстве графики в фильмах и играх.
2. Методы, задействующие алгоритмы машинного обучения оптимизации движения. Например, метод Pictorial structure framework, основанный на модели Случайного леса [6]. Минусом данных моделей является то, что они требуют присутствия четко видимых частей тела.
3. Решения на основе моделей глубокого обучения. Начиная с 2014 года [7], исследователи успешно применяют различные архитектуры глубоких нейронных сетей, например, сверточных, для решения задачи определения позы. В данной работе рассматриваются именно такие методы.

Одним из факторов, которые усложняют поставленную задачу является то, что на изображении может быть представлено несколько объектов (людей/животных). Существует 2 подхода решения этой проблемы:

1. **Сверху вниз (top-down)** - сначала происходит обнаружение объекта целиком, а затем система разбивает объект на более мелкие компоненты - ключевые точки позы. Этот подход эффективен при обработке изображений с большим количеством объектов, однако плохо работает при перекрытиях и сложных позах.
2. **Снизу вверх (bottom-up)** - сначала происходит обнаружение отдельных частей объекта - ключевых точек позы человека, а затем система объединяет эти части для определения общей позы объекта. Этот подход решает проблему сложных поз и позволяет более точно определять позу. Однако требует больше вычислительных ресурсов, чем предыдущий и может быть менее эффективным, когда объектов много.

В статье [8] был предложен метод DeepCut на основе методики “снизу вверх”. Он использует набор сверточных нейронных сетей для распознавания всех частей тела, их разметки и разделения на каждого человека.

В статье [9] была представлена OpenPose - первая система реального времени с открытым исходным кодом для определения двумерных поз нескольких человек на изображении. Предложенная модель использовала метод “снизу вверх”. OpenPose состоит из VGG-19 для извлечения из изображения и двух ветвей сверточных нейронных сетей (рисунок 3).

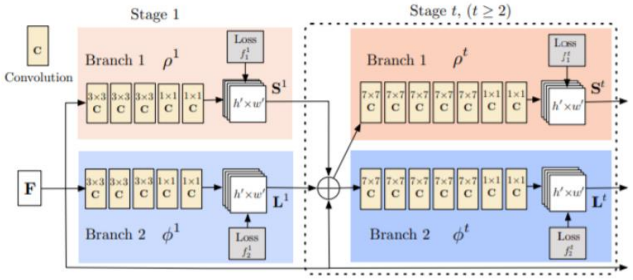


Рис 3. Архитектура OpenPose

Работа [10] описывает нейронную сеть HRNet, которая может использоваться в широком спектре задач визуального распознавания, в том числе в задаче распознавания позы человека по методу “снизу вверх”. Основное нововведение статьи заключается в том, что нейронная сеть не кодирует входное изображение как представление с низким разрешением, например, с помощью ResNet, VGG. Вместо этого нейронная сеть поддерживает представления с высоким разрешением на протяжении всего процесса (рисунок 4).

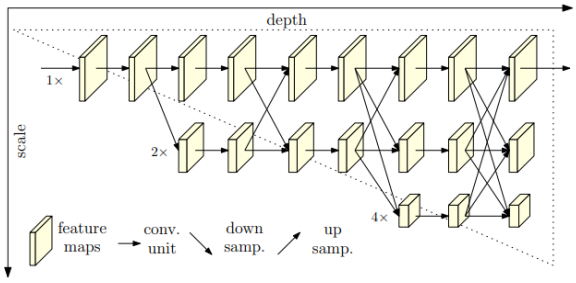


Рис 4. Ахитектура сети HRNet

В 2023 году была разработана система RTMPose [11], которая предложила решение (метод “снизу вверх”), которое не только показывает хорошую точность определения позы, но и уменьшает задержку, позволяя использовать его в real-time приложениях. Архитектура модели показана на рисунке 5.

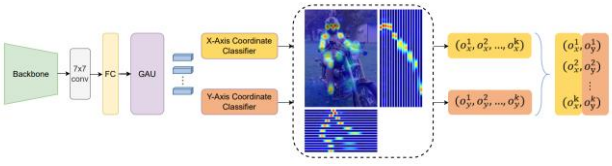


Рис 5. Архитектура RTMPose

Системы HRNet и RTMPose внутри обе используют

сеть Faster RCNN для точной детекции объектов в реальном времени. Эта сеть, описанная в статье [12], объединяет сети Fast RCNN [13] и SPP-net [14].

III. Данные

A. *Описание данных*

Наиболее распространенным форматом данных являются датасеты с изображениями и размеченными на них ключевыми точками. Ключевые точки могут обозначать не только позу тела человека, но и позу животных, точки на лице, кисти рук и т.п. (рисунок 6).

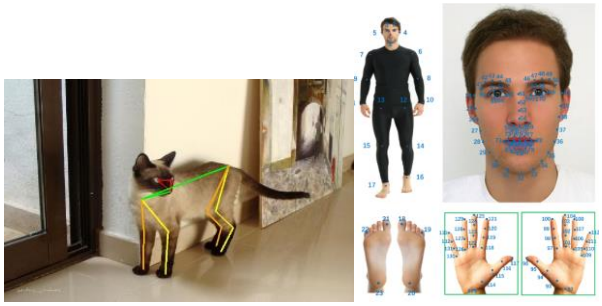


Рис. 6. Пример разметки ключевых точек объектов

Существует большое количество датасетов, предназначенные для разных задач. Среди них COCO, uBody, Face2d, Hand2d AP-10k. Также существуют объединенные датасеты, например, Face6, Cocktail14, Hand5, Body8.

Также существуют и датасеты для оценки позы в трехмерном пространстве: 3DPW, Waymo. RICH и другие (рисунок 7).

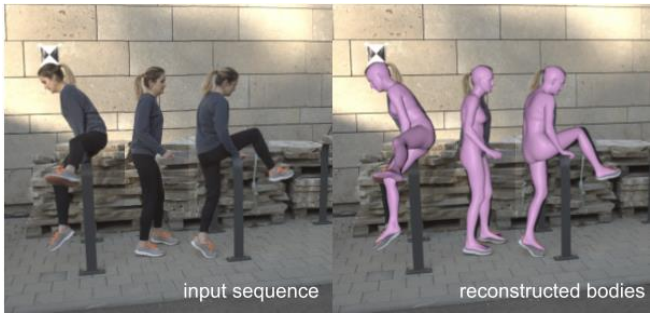


Рис. 7. Пример датасета с трехмерным представлением позы

Также при создании датасетов нередко используются и синтетические данные. В частности, в датасете InfiniteRep используют сгенерированные сцены людей, выполняющих упражнения в разных обстановках и освещении (рисунок 8).

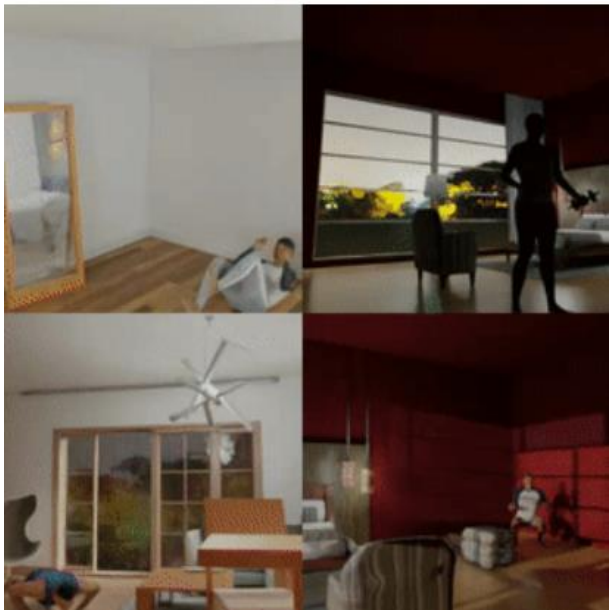


Рис. 8. Пример синтетических данных для оценки поз

IV. Проведенные исследования и результаты

A. *Предобработка датасета*

В качестве датасета для обучения нейросетей использовалась подвыборка из датасета COCO размером 97 изображений.

Датасет был разделен на обучающую и тестовую выборки в соотношении 80%-20% с перемешиванием.

B. *Выбор метрик*

При обучении происходила минимизация целевой функции KLDiscretLoss при помощи оптимизатора Adam.

$$KLDiscretLoss = \sum_{i} P(i) ln(\frac{P(i)}{Q(i)})$$

где $P(i)$ - вероятность истинного распределения i позиции тепловой карты, $Q(i)$ - вероятность предсказанного распределения.

В качестве метрики для оценки качества распознавания поз использовались PCKAccuracy.

$$PCKAccuracy = \frac{1}{N}\sum_{i=1}^{N} 1(||x_i^{pred} - x_i^{true}|| \leq \alpha L)$$

где $||x_i^{pred} - x_i^{true}||$ - евклидово расстояние между предсказанной и истинной позицией ключевой точки i, α - порог, определяемый как доля от эталонного размера L, $1()$ - индикаторная функция, которая равна 1, если условие верно, и 0 в противном случае.

C. *Использованные модели машинного обучения*

RTMPose - Real-Time Multi-Person Pose Estimation - это сеть, оптимизированная для быстрой и точной детекции поз нескольких людей. Она использует легковесные архитектуры, такие как MobileNet или EfficientNet, для базовых фичей и специализированные модули для обработки позы, включая улучшенные тепловые карты и адаптивные методы нелинейной регрессии, что позволяет достигать высокой производительности и точности даже на устройствах с ограниченными вычислительными ресурсами.

HRNet - High-Resolution Network (HRNet) для детекции поз использует идею поддержания высокого разрешения на всех уровнях обработки изображения, что позволяет сохранить детализированную информацию о позе и улучшить точность локализации ключевых точек.

Faster RCNN - сеть для детекции рамок. Поскольку обе сети используются в архитектуре сверху вниз, отдельно найти рамки, в которых находится человек, а после в каждой рамке произвести детекцию позы

Все представленные архитектуры нейронных сетей были реализованы с помощью библиотек mmpose [15] и mmdet [16].

D. *Результаты обучения и инференса*

Ввиду ограниченности вычислительных ресурсов, были использованы модификации нейросетей небольших размеров (rtmpose - medium, hrnet - w48), а их обучение происходило на 40 эпохах. В результате были получены следующие значения метрики PCKAcc, показанные в Таблице 1. Также рассчитаны метрики для датасетов COCO+ubody (350 тыс. изображений) и AP-10k (10 тыс. изображений).

Модель	AP-10k AP	COCO + ubody AP	COCO tiny PCKAcc
HRNet	0.728	0.594	0.457155
RTMPose	0.722	0.606	0.650190

ТАБЛИЦА I. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

На рисунке 9 изображение примеры распознавания на тестовой выборке.

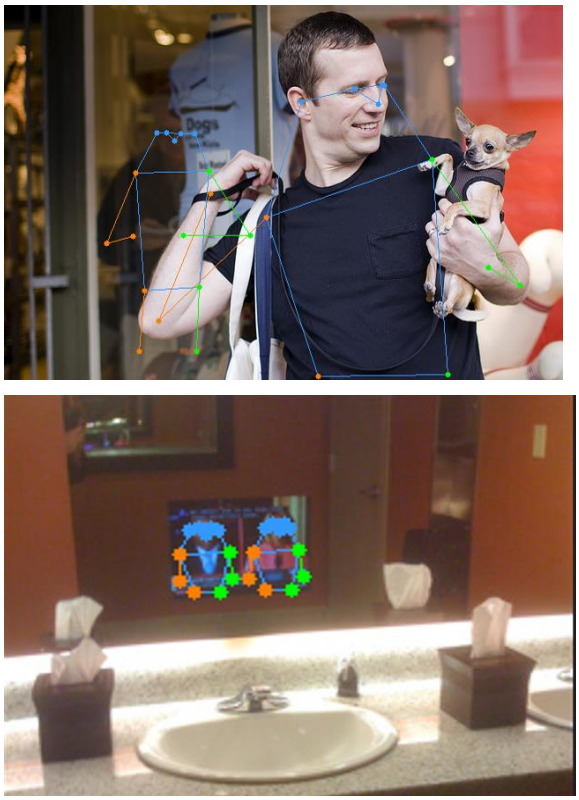


Рис. 9. Результаты распознавания поз на тестовой части датасета при помощи RTMPose

На рисунке 10 представлены примеры распознавания поз на собственных изображениях не из обучающей и тестовой выборки.

Рис. 10. Результаты распознавания поз на пользовательских изображениях при помощи RTMPose

V. Заключение

В рамках данной работы был проведен анализ литературы, посвященной определению поз людей на изображении. Для исследования были взяты 2 нейронные сети: HRNet и RTMPose. Они были протестированы на датасетах с изображениями людей: COCO + ubody и COCO tiny, а также на датасете AP-10K с изображениями животных.

По результатам тестирования видно, что модель RTMPose имеет лучшую точность на датасетах с изображениями людей, однако немного проигрывает в точности при определении позы животных. RTMPose показала на маленькой обучающей выборке заметно лучший результат, по сравнению с HRNet (0.65 против 0.457), однако на больших датасетах разницы почти нет. Это говорит о том, что RTMPose быстрее обучается и обладает лучшей обобщающей способностью.

Подводя итог, можно с уверенностью говорить об успешном приложении нейронных сетей к задаче определения позы человека или нескольких людей на изображении. Методы, исследованные в данной работе обеспечивают хорошее качество распознавания, а также, что не менее важно, позволяют использовать их в приложениях в реальном времени.

Список литературы

[1] Bhuyan, K.; Van Westen, C.; Wang, J.; Meena, S.R. “Mapping and characterising buildings for flood exposure analysis using open-source data and artificial intelligence”, Nat. Hazards, vol 119. pp 1-31.

[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] S. H. Zabihifar, A. N. Semochkin, E. V. Seliverstova, and A. R. Efimov, “Unreal mask: one-shot multi-object class-based pose estimation for robotic manipulation using keypoints with a synthetic dataset,” Neural Computing and Applications, vol 33, Oct 2021, pp. 12283–12300, doi: 10.1007/s00521-020-05644-6.

[4] Y. S. Chernyshova, B. I. Savelyev, S. V. Solodov, S. V. Pronichkin, “Applying distributed ledger technologies in megacities to face anthropogenic burden challenges,” in IOP Conference Series: Earth and Environmental Science, 2022, vol. 1069, no. 1. doi:10.1088/1755-1315/1069/1/012028.

[5] 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.

[6] «Shrinkage Optimized Directed Information using Pictorial Structures for Action Recognition» Chen X., et al. // Arxiv. URL: https://arxiv.org/abs/1404.3312 (доступ 10.05.2024)

[7] A. Toshev, C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks // 2014 IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 1653–1660.

[8] Rajchl M. et al. DeepCut: Object Segmentation From Bounding Box Annotations Using Convolutional Neural Networks // IEEE Transactions on Medical Imaging. – 2017. –V. 36. – N. 2. – P. 674–683.

[9] Cao Z., rt al. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. –V. 43. – N. 1. – P. 172–186.

[10] Wang J., et al. Deep High-Resolution Representation Learning for Visual Recognition // IEEE Transactions on Pattern Analysis & Machine Intelligence. – 2021. –V. 43. – N. 10. – P. 3349–3364.

[11] «RTMPose: Real-Time Multi-Person Pose Estimation based on MMPose» Jiang T., et al. // Arxiv. URL: https://arxiv.org/abs/2303.07399 (доступ 10.05.2024)

[12] Ren S., et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // IEEE Transactions on Pattern Analysis & Machine Intelligence. – 2017. –V. 39. – N. 6. – P. 1137–1149.

[13] Girshick R. Fast R-CNN // 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile. – 2015. – P. 1440–1448.

[14] He K.,et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition // Computer Vision – ECCV 2014. – 2014. – V. 8691.

[15] MMPose python library // Github. URL: https://github.com/open-mmlab/mmpose (доступ 10.05.2024)

[16] MMDetection python library // Github. URL: https://github.com/open-mmlab/mmdetection (доступ 10.05.2024)

ГРНТИ 28.23.37
ASJC 1706
OECD 01.02.EP

Количественный и качественный анализ аудитории Telegram в разрезе рекомендаций с использованием больших языковых моделей

А. В. Соседка
кафедра инженерной кибернетики
НИТУ МИСИС
Москва, Россия
m1801239@edu.misis.ru

П. И. Ибрагимов
кафедра инженерной кибернетики
НИТУ МИСИС
Москва, Россия
peter.ibragimov@gmail.com

***Аннотация* — В работе рассматривается применение больших языковых моделей для семантического анализа сообщений в массиве каналов и последующий анализ полученных данных, кластеризация и создание классификатора с использованием нейронных сетей.**

Ключевые слова — Большие языковые модели, Telegram, RNN, Коллаборативная фильтрация, Тематическое моделирование

I. Введение

В результате стремительного развития средств коммуникации такие платформы как Telegram стали основным инструментом для распространения информации с использованием такого инструмента как публичные каналы. Пользователи могут присоединяться к публичным каналам и получать релевантную информацию по тематике канала, например новости в сфере научных открытий или анонсы мероприятий в студенческих сообществах.

До недавнего времени каждый публичный канал в Telegram был изолирован ввиду отсутствия явной связанности между ними. Но в обновлении от 30 ноября 2023 года появился функционал “Похожие каналы”, который предоставляет список рекомендуемых каналов, основываясь на коллаборативной фильтрации [1]. С помощью этого можно собрать анонимный ориентированный граф взаимодействий пользователей с каналами, где будут взвешенные рёбра между каналами (узлами), обозначающие ранг конечной вершины относительно начальной.

Одним из основных прорывов в области обработки естественного языка является появление больших языковых моделей [2] основанных на архитектуре трансформеров [3]. На сегодняшний день [4] они позволяют получать семантическую информацию из корпусов текста и предоставляют информацию из текста в векторной формате, облегчая дальнейшую работу с информацией для последующей обработки и анализа.

Развитие методов глубокого обучения [5], широкое распространение технологий искусственного интеллекта [6,7] и последние инновации в сфере больших языковых моделей позволяют проводить количественный и качественный анализ каналов Telegram, учитывая сообщения и связи между ними. Используя текстовую информацию из сообщений каналов, а именно векторное представление корпуса текста, полученную посредством кодирования предложений [8] можно обучить классификатор для создания рекомендательной системы, которая будет предлагать пользователю каналы как на основе содержания, так и на основе коллаборативной фильтрации.

В качестве основной задачи выдвигается кластеризация, основываясь только на структурной информации с графа, а также тематическое моделирование каналов на основании сообщений. Разработка программного комплекса выполняется на языке программирования Python [9].

II. Наборы данных

Датасеты, которые применялись в данной статье, были собраны с помощью разработанного автоматизированного парсера Telegram.

Telegram канал является способом коммуникации, аналогичный ведению блога в таких социальных сетях, как Twitter, VK, Youtube. Основным отличием от указанных сетей является то, что, по умолчанию, Telegram каналы псевдоанонимны, а связь между автором и пользователями односторонняя. Если автор канала вручную не активирует возможность писать комментарии, а также не укажет контактные данные для связи с собой, пользователи не смогут как-либо с ним связаться. Важно отметить, что авторы могут как размешать текстовый публикации, например новости, анекдоты, так и видео, аудио файлы и другие.

Каждый достаточно большой публичный канал, то есть тот, начать читать который может любой желающий, имеет рекомендательную связку с другими каналами. Данная связка создается самим Telegram и, исходя из документации к API, представляет собой коллаборативную рекомендательную систему. Принцип действия тривиален - Telegram рекомендует те каналы, на которые чаще всего подписываются подписчики анализируемого аккаунта. Важно отметить, что Telegram фильтрует рекомендации к каналам, на которые пользователь уже подписан, поэтому в данной работе был использован новый, не подписанный ни на какие каналы, аккаунт.

Telegram, представляя собой классический клиент-серверный проект, имеет API для взаимодействия между сервером и клиентом. Данный интерфейс использует

открытый протокол MTProto [10], а методы и способы взаимодействия с серверной частью Telegram являются открытыми и доступными. Таким образом был разработан сервис, который по команде пользователя мог собрать данные с интересующих каналов и представить их в нужном формате.

Указанный сервис принимает на вход 3 аргумента: название канала, с которого надо начать обход, глубина обхода и максимальное количество соединений канала, которые надо проанализировать. Сервис выгружает корневой канал, последние 80 постов в указанном канале, рекомендации для канала, далее данный процесс рекурсивно повторяется для рекомендаций канала. Каждый датасет состоит из 3 частей, описание каждой представлено в таблицах 1 – 3.

ТАБЛИЦА I. Описание таблицы channel

Название поля	Тип данных	Назначение
tg_id	int64	Уникальный идентификатор канала в Telegram
tg_title	string	Название канала
tg_username	string	Именованный идентификатор канала
uploaded_at	timestamp	Дата загрузки канала в БД
participant_count	int	Количество подписчиков канала
tg_about	string	Описание канала
is_leaf	bool	Является ли этот узел конечным в графе рекомендаций каналов

ТАБЛИЦА II. Описание таблицы channel_edge

Название поля	Тип данных	Назначение
tg_id_in	int64	Идентификатор начального узла ребра
tg_id_out	int64	Идентификатор конечного узла ребра
order	int64	Вес ребра. Чем ниже - тем выше находится ребро в рекомендациях Telegram

ТАБЛИЦА III. Описание таблицы message

Название поля	Тип данных	Назначение
tg_id	int	Уникальный идентификатор сообщения в Telegram
text	string	Текст сообщения
tg_chat_id	int64	Идентификатор чата, в котором было отправлено сообщение

В результате работы сервиса было собрано 3 датасета: корневыми каналами выступили две официальные площадки разных вузов и один новостной канал. Все 3 датасета собирались с глубиной обхода графа 3, максимальная исходящая степень каждой вершины - 20. Сводная статистика по датасетам представлена в таблице 4.

ТАБЛИЦА IV. Статистика собранных датасетов

Таблица	Размер в записях
Официальная площадка вуза 1	
channel	9186
channel_edge	33791
message	266136
Официальная площадка вуза 2	
channel	7608
channel_edge	32278
message	249929
Новостной канал	
channel	2669
channel_edge	16998
message	163918

Разница в количестве записей в датасетах связана с разной степенью связанности: например, у канала А может быть рекомендации Б, В, Д, а у канала Б - А, В, Д, это приводит к тому, что будет выгружено всего 4 канала: А, Б, В, Д, даже при глубине рекурсии 2.

III. Используемые инструменты

A. *Sentence Transformers*

В работе [8] авторы модифицируют архитектуру BERT [11] с помощью Siamese Net [12], тем самым позволив получать векторное представление целого корпуса текста, нежели каждого слова по отдельности. Архитектура их решения представлена на рисунке 1. Это позволяет использовать кодировать семантическое значение предложений в контексте окружающих слов. В данной работе было принято решение использовать одну из мультиязыковых моделей [13] для векторного кодирования текстового составляющего сообщений в Telegram каналах, а именно *paraphrase-multilingual-mpnet-base-v2.*

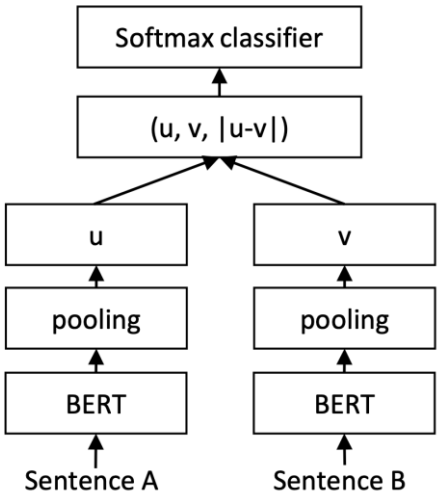


Рис. 1. Архитектура SBERT модели при обучении на задаче классификации. Чтобы получить нужные векторные представления предложения, подаётся только одно предложение и берётся вектор u

B. BERTopic

Для решения задачи тематического моделирования [14] в данной работе предполагается использование методов из библиотеки BERTopic [15], которые предоставляют удобный интерфейс и готовый пайплайн (рисунок 2) для обработки текстовых документов и визуализации результатов моделирования. С помощью этого инструмента, возможно кластеризовать каналы, используя семантические представления сообщений, и в итоге получить кластера каналов по тематике, которые будут отличными от кластеров, полученных только с помощью структурной информации с графа.

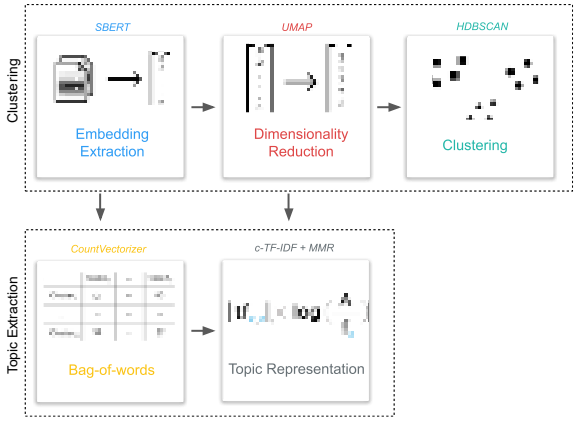


Рис. 2. Основные этапы тематического моделирования: перевод текста в векторное представление, понижение размерности полученных векторов, кластеризация векторов, токенизация текстовых корпусов и выборка ключевых слов из документов посредством c-TF-IDF

C. NetworKit

В качестве основной библиотеки для работы с графовым представлением данных была выбрана NetworKit [16] ввиду быстродействия, поддержки широкого спектра алгоритмов и наличия интерфейса для взаимодействия на языке Python. В частности, предполагается использование встроенного алгоритма *Parallel Louvain Method* [17] для определения сообществ в графе, что позволит получить структурное представление каналов и их взаимодействие между собой.

D. GRU-based классификатор

В качестве модели для классификатора используется архитектура сиамской модели (рисунок 3), где в качестве энкодера используется модель из sentence-transformer *(paraphrase-multilingual-mpnet-base-v2)*, которая предоставляет векторное представление для текста из сообщений из канала, которые потом подаются последовательно в две параллельные GRU [18] ячейки, одна из которых получает векторное представление сообщения, а другая - относительный номер сообщения в последовательности, то есть, для самого первого сообщения в последовательности будет порядковый номер 20, так как берутся последние 20 сообщений для каждого канала, а у самого последнего - 1.

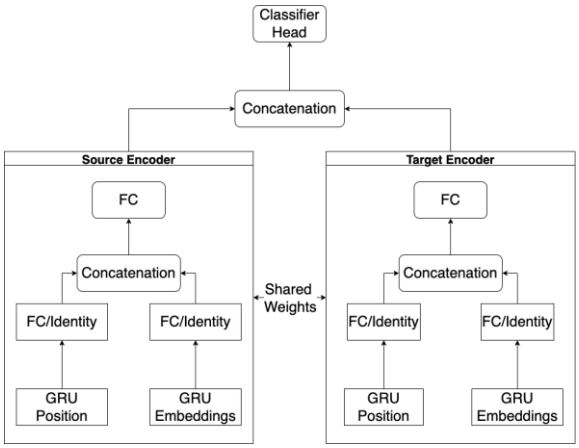


Рис. 3. Архитектура предлагаемой модели

Используя двойное кодирование информации (последовательное и текстовое), можно получить общее представление о канале, учитывая семантику его сообщений. При использовании архитектуру сиамской модели предполагается общая матрица весов между двумя параллельными моделями. На выходе из двух GRU модулей вектора проходят через многослойный персептрон и конкатенируются, итоговый вектор подаётся на классифицирующий модуль, также представляющий из себя многослойный персептрон с двумя выходами, которые определяют два класса (бинарная классификация) - является ли канал А рекомендуемым для канала Б.

IV. Результаты

A. Поиск сообществ с помощью Parallel Louvain Method

В ходе работы было был собран общий граф по трём датасетам (два университетских и один новостной). С помощью алгоритма *Parallel Louvain Method* было найдено 248 сообществ (таблица 5), используя стандартные гиперпараметры *gamma=1.0*. Графическое представление графа с выделенными сообществами (рисунок 4) позволяет увидеть несколько сильно связанных компонент, которые образуются вокруг корневых вершин (каналов, которые являлись начальными при сборе датасетов). При понижении параметра gamma уменьшалось количество найденных

сообществ, но исключительно за счёт повышения размера самых больших сообществ, проблема маленьких сообществ размером в несколько каналов не решилась, скорее всего из-за выбросов в рекомендуемых каналах от Telegram.

ТАБЛИЦА V. Описание результатов работы алгоритма Parallel Louvain Method

Количество найденных сообществ	248
Минимальный размер сообщества	10
Максимальный размер сообщества	1016
Средний размер сообщества	54.5484
Дисбаланс	18.4727
Разрезанных рёбер	1019
Модулярность	0.86063

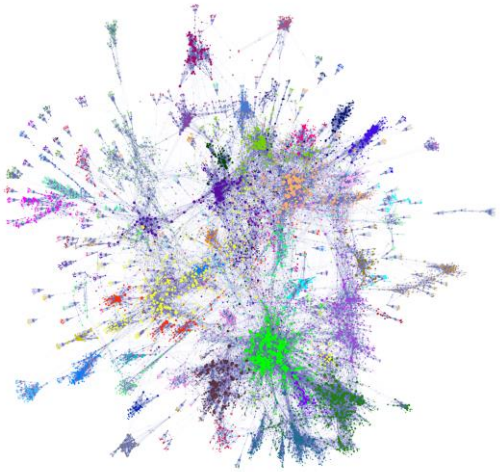


Рис. 4. Отображение итогового графа. Вершины раскрашены в цвета сообществ, в которых они состоят

B. *Тематическое моделирование*

Для получения тематик каналов были взяты последние сообщений из каждого канала и использованы как документы для тематического моделирования. В качестве энкодера для получения векторного представления сообщений использовалась модель *paraphrase-multilingual-mpnet-base-v2*. Ввиду большого количества каналов и сообщений, здесь и далее были рассмотрены три датасета по отдельности.

Были выбраны темы *новости, политика, образование, наука, развлечение, технологии* и *блоги* для кластеризации относительно семантики сообщений в канале. После отображения (рисунок 5) явно видно, что темы *образование* и *наука* располагаются относительно близко структурно так же, как *новости* и *политика.* Из-за специфики собранных данных тематика *блоги* и *развлечение* находятся на окраинах графа, так как они наиболее удалены от тематик стартовых вершин трёх датасетов.

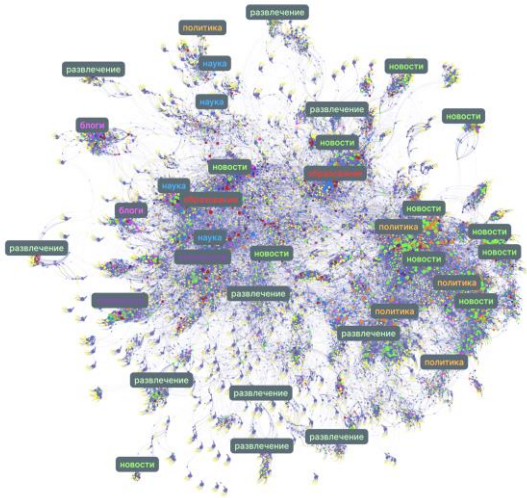


Рис. 5. Отображение графа, размеченного на отдельные тематики

C. *Обучение классификатора*

Для обучения классификатора необходимо получить общее векторное представление канала, основываясь на его последних сообщениях. При обучении классификатора было произведено выравнивание относительно коллаборативной фильтраций, то есть в качестве таргетов были выбраны существующие рекомендации от Telegram. При обучении оптимизировалась объединённая функция ошибки Binary Cross Entropy (частный случай log-loss) [19] для решения задачи бинарной классификации

$$\mathcal{L}(y,p) = -(y\log(p) + (1-y)\log(1-p)) \quad (1)$$

и Triplet Margin Loss [20]

$$\mathcal{L}(A,P,N) = max(|f(A)-f(P)|_2 - |f(A)-f(N)|_2 + \alpha, 0) \quad (2)$$

для распределения векторов каналов в пространстве так, что близкие друг к другу каналы рекомендовали чаще, чем дальние. В качестве архитектуры была выбрана сиамская модель, которая включает в себя две модели с общими весами для кодирования каналов в векторное пространство, используя последовательность векторных представлений сообщений в канале и последовательность позиций этих сообщений (positions и embeddings соответственно) и отдельный полносвязный слой для классификации. Было выбрано два варианта моделей, одна имеет дополнительный полносвязный слой на выходе между GRU модулем и конкатенацией, а другая нет. Модели обучались 15 эпох на каждом датасете отдельно, каждый датасет был разбит на три выборки (обучающая, валидирующая и тестовая) в отношении 8:1:1. В качестве метрик были выбраны *Accuracy*, *Precision*, *Recall*, и *F1-мера.* TP (True Positive) означает верную классификацию каналов (канал B является рекомендуемым для канала A), FP (False Positive) означает ложноположительную классификацию (канал B неверно рекомендуется каналу A), TN (True Negative) означает негативную классификацию (канал B верно не рекомендуется для канала A), а FN (False

Negative) означает ложную негативную классификацию (канал B ложно рекомендуется для канала A)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN} \quad (6)$$

ТАБЛИЦА VI. Метрики по тестовой выборке для модели с дополнительным полносвязным слоем

Датасет	Модель с FC			
	Accuracy	Precision	Recall	F1
news	0.691	0.6958	0.6891	0.6925
uni1	0.6241	0.5718	0.6386	0.6034
uni2	0.6339	0.5865	0.648	0.6157

ТАБЛИЦА VII. Метрики по тестовой выборке для модели без дополнительного полносвязного слоя

Датасет	Модель без FC			
	Accuracy	Precision	Recall	F1
news	0.7049	0.7139	0.7012	0.7075
uni1	0.6318	0.6782	0.6206	0.6481
uni2	0.6395	0.6677	0.632	0.6494

Как видно из таблицы 6, использование дополнительного полносвязного слоя не позволяет модели полноценно сойтись, из-за чего она показала худшие результаты, чем альтернативный вариант, таблица 7. Результаты модели можно улучшить, увеличив время обучения и количество данных для обучения. Так же для улучшения качества метрик можно изменить параметры модели, изменив GRU энкодер на Transformer и добавив тем самым механизм внимания, или доразметить датасет, используя обратную связь от пользователей, какие каналы им нравятся больше.

V. Заключение

Был собран анонимизированный датасет из открытых источников Telegram для структурного и семантического анализа каналов массового вещания и их смежной аудитории, которая выражается в ранжированных рекомендациях коллаборативной фильтрации. Приведены различные варианты анализа данного датасета, включающие как анализ графа и его структуры, так и семантической информации из сообщений в канале Telegram.

В данной работе проведён глубокий анализ аудитории Telegram, основанный на использовании больших языковых моделей для семантического разбора текстов из публичных каналов. Это позволило классифицировать каналы и кластеризировать их по темам, что значительно упрощает дальнейший анализ аудитории. Благодаря разработанному подходу, основанному на структурном анализе и семантическом моделировании, был обучен классификатор, достигающий 70% точности на тестовой выборке, который предлагает каналы пользователям не только на основе коллаборативной фильтрации, но и с учётом содержания сообщений в каналах.

Литература

[1] Schafer J. B. et al. Collaborative filtering recommender systems //The adaptive web: methods and strategies of web personalization. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2007. – С. 291-324.

[2] Radford A. et al. Improving language understanding by generative pre-training. – 2018.

[3] Vaswani A. et al. Attention is all you need //Advances in neural information processing systems. – 2017. – Т. 30.

[4] Minaee S. et al. Large language models: A survey //arXiv preprint arXiv:2402.06196. – 2024.

[5] Chernov T. S. et al. Image quality assessment for video stream recognition systems //Tenth International Conference on Machine Vision (ICMV 2017). – SPIE, 2018. – Т. 10696. – С. 473-480.

[6] Yakovlev A. A., Kondybayeva A. B., Solodov S. V. Intelligent System for Collecting, Analyzing and Managing Data in the Field of Medicine //2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). – IEEE, 2019. – С. 1-6.

[7] Анохин К. В. и др. Искусственный интеллект для науки и наука для искусственного интеллекта //Вопросы философии. – 2022. – Т. 3. – С. 93-105.

[8] Reimers N., Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks //arXiv preprint arXiv:1908.10084. – 2019] и модель основанную на RNN [Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors //nature. – 1986. – Т. 323. – №. 6088. – С. 533-536..

[9] Широков А.И., Пышняк М.О. Информатика. Основные понятия теории. Разработка программ на языке программирования Питон. Основные понятия. Часть 1. Учебник. Издательский Дом НИТУ МИСиС, 2020.- 142 с.

[10] Lee J. et al. Security analysis of end-to-end encryption in Telegram //Simposio en Criptografía Seguridad Informática, Naha, Japón. Disponible en https://bit. ly/36aX3TK. – 2017.

[11] Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding //arXiv preprint arXiv:1810.04805. – 2018.

[12] Chicco D. Siamese neural networks: An overview //Artificial neural networks. – 2021. – С. 73-94.

[13] Reimers N., Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation //arXiv preprint arXiv:2004.09813. – 2020.

[14] Vayansky I., Kumar S. A. P. A review of topic modeling methods //Information Systems. – 2020. – Т. 94. – С. 101582.

[15] Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure //arXiv preprint arXiv:2203.05794. – 2022

[16] Angriman E. et al. Algorithms for large-scale network analysis and the NetworKit toolkit //Algorithms for Big Data: DFG Priority Program 1736. – Cham : Springer Nature Switzerland, 2023. – С. 3-20.

[17] Staudt C. L., Meyerhenke H. Engineering parallel algorithms for community detection in massive networks //IEEE Transactions on Parallel and Distributed Systems. – 2015. – Т. 27. – №. 1. – С. 171-184.

[18] Cho K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation //arXiv preprint arXiv:1406.1078. – 2014

[19] Good I. J. Rational decisions //Journal of the Royal Statistical Society: Series B (Methodological). – 1952. – Т. 14. – №. 1. – С. 107-114.

[20] Schultz M., Joachims T. Learning a distance metric from relative comparisons //Advances in neural information processing systems. – 2003. – Т. 16.

ГРНТИ 28.23.15, 28.23.37
ASJC 1702, 1707
OECD 01.02.EP

Методы глубокого обучения для обнаружения огня

А. А. Ступина
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
stupinaaa99@gmail.com

***Аннотация* — в данной работе проводится сравнение эффективности двух различных подходов к распознаванию огня на изображениях: использование архитектуры сверточной нейронной сети ResNet50 и применение модели FireNet. Подробно рассматриваются архитектуры обеих моделей, их особенности и принципы работы. Исследование демонстрирует, как выбор архитектуры нейросети может влиять на качество решения задачи классификации изображений. Анализ преимуществ и недостатков каждого подхода предоставляет важные данные для разработки и оптимизации систем обнаружения огня.**

Ключевые слова — компьютерное зрение, обнаружение огня, классификация изображений, ResNet50, FireNet

I. Введение

Обнаружение огня на изображениях является критически важной задачей для множества приложений, включая системы безопасности, мониторинг лесов, управление в чрезвычайных ситуациях и автоматизированное видеонаблюдение. Современные технологии позволяют использовать алгоритмы компьютерного зрения для раннего обнаружения огня, что может существенно снизить ущерб от пожаров, обеспечивая быстрое и точное реагирование.

В последние годы значительные усилия исследователей были направлены на разработку и улучшение алгоритмов машинного обучения [1, 2], способных эффективно классифицировать изображения по наличию на них огня. Наиболее перспективными в этом направлении являются сверточные нейронные сети (CNN), которые показали выдающиеся результаты в задачах компьютерного зрения благодаря своей способности извлекать сложные признаки из изображений [3].

В большинстве существующих подходов к обнаружению огня с использованием компьютерного зрения и сверточных нейросетей применяется тонкая настройка таких известных архитектур, как VGG16 [4], ResNet [5], GoogleNet [6] и другие [7]. Эти модели демонстрируют высокую эффективность в многочисленных задачах компьютерного зрения благодаря своей способности к глубокому и сложному анализу изображений. Однако при их использовании в реальном времени на устройствах с ограниченными вычислительными ресурсами возникают значительные трудности [8]. Большой размер этих моделей и их многослойная структура требуют значительных вычислительных мощностей.

Авторы нейросети FireNet [8] предложили решение этих проблем, разработав легковесную архитектуру, специально оптимизированную для мобильных и встроенных приложений. Модель не только значительно меньше по размеру и проще по сложности, но и способна эффективно работать в условиях ограниченной вычислительной мощности. Как заявляют создатели, FireNet демонстрирует отличные результаты в задачах обнаружения огня в реальном времени, поддерживая частоту обработки более 24 кадров в секунду даже на недорогих одноплатных компьютерах, таких как Raspberry Pi 3B [9]. Таким образом, модель обеспечивает необходимую производительность при значительно меньшем потреблении ресурсов и, соответственно, при меньших затратах.

В данной работе проведено сравнение качества решения задачи классификации архитектур FireNet и ResNet. Целью сравнения является оценка того, удалось ли разработчикам FireNet достичь улучшения производительности без ущерба для точности классификации. Это позволит не только оценить практическую применимость FireNet в реальных условиях, но и даст важное понимание о том, можно ли сократить ресурсоемкость нейросетевых решений без значительной потери в качестве работы.

II. Наборы данных

Для обучения и тестирования рассматриваемых в данной работе нейросетей использовались открытые наборы данных. Рассмотрим их.

A. Дополненный датасет Foggia's

Этот датасет [10] использовался для обучения сети FireNet, а также для дообучения модели на основе ResNet50 [5]. Этот датасет содержит 31 видеофайл с изображениями огня, а также изображениями, на которых он отсутствует. Несмотря на большое количество итоговых изображений, минус данного датасета заключается в том, что он содержит множество похожих и повторяющихся кадров. В связи с этим авторы FireNet дополнили [8] его большим количеством изображений, широко различающихся по месту действия, времени суток, цветовой палитре и пр. Примеры изображений можно увидеть на рисунке 1.

B. ImageNet

ImageNet [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.

Рис. 1. Примеры классов изображений: а, б) Огонь, в) Отсутствие огня

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. Перенос обучения позволяет значительно ускорить процесс обучения и улучшить производительность моделей на конкретных задачах.

C. Открытые наборы данных

Для сравнения эффективности двух нейросетевых архитектур были выбраны два открытых датасета, один из которых авторы статьи [8, 10] использовали для тестирования своей модели. Он содержит 871 изображение огня в различных условиях, в т.ч. изображения с телефонных камер и камер видеонаблюдения. Такой подход обусловлен необходимостью проверить работу моделей в реальных условиях, где источники изображений часто являются непрофессиональными и имеют вариативное качество, различное освещение и углы съемки.

В качестве второго набора данных использовалась комбинация открытых датасетов Fire Image Dataset [12] и Non Fire Image Dataset [13], суммарно содержащая 9553 изображения. Итоговый датасет включает в себя не только изображения с огнем, но и большое количество изображений, на которых огня нет, но которые очень схожи по освещению и цветотону. Такие изображения могут содержать кадры с закатами, светом от ламп, отражениями и другими источниками света, которые визуально напоминают огонь. Этот набор данных позволяет тщательно оценить, насколько хорошо модель может отличать настоящий огонь от других схожих объектов, минимизируя количество ложных срабатываний. Примеры изображений представлены на рисунке 2.

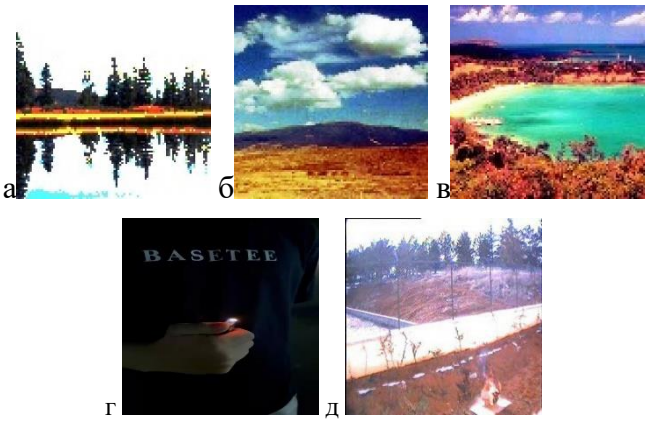


Рис. 2. Примеры сложных для классификации изображений: а, б, в) Схожесть по цветотону и текстуре с пламенем; г, д) Сложно отличимое пламя

III. Нейросетевые архитектуры

A. FireNet

В работе [8] решается задача классификации изображений с огнем для улучшения функционирования систем обнаружения пожаров. В перспективе такие системы должны стать частью умных городов [14, 15]. Это важно, так как физические детекторы, такие как температурные датчики и датчики дыма, могут быть недостаточно надежны и часто вызывают ложные срабатывания по различным причинам, например, из-за курения или других источников дыма и тепла. На рисунке 3 представлена полная схема архитектуры предложенной сети, где видно, что FireNet состоит из 14 слоёв, включая пулинговые слои, дропаут и выходной слой с функцией активации Softmax. Сеть имеет три свёрточных слоя, каждый из которых сочетается с пулинговым слоем и слоем дропаута. Все эти слои используют функцию активации Rectified Linear Unit (ReLU), за исключением последнего слоя, где используется Softmax. Общее количество обучаемых параметров составляет 646,818 (размер на диске примерно 7,45 МБ).

Первый слой — это свёрточный слой, который обрабатывает цветное изображение размером (64×64×3). Этот размер выбран на основе эмпирических результатов сравнения различных размеров. Размер входных данных можно увеличить до (128×128×3) без существенного влияния на количество кадров в секунду (FPS). Этот слой содержит 16 фильтров с размером ядра (3, 3).

В каждом из двух последующих свёрточных слоёв удваивается количество входных признаков при постоянном размере ядра. За этими слоями следуют flatten слой и два полносвязных слоя по 256 и 128 нейронов соответственно. Заключительный слой — полносвязный слой слой с двумя нейронами, который выполняет функцию выходного слоя предсказания.

B. ResNet50

ResNet50 — мощная модель классификации изображений, которая может быть обучена на больших наборах данных (в частности, ImageNet) и достигать передовых результатов [16, 17]. Одной из ключевых инноваций является использование остаточных соединений, которые позволяют сети обучаться на множестве остаточных функций, отображающих входные данные в желаемый

результат. Эти остаточные соединения позволяют сети обучаться на гораздо более глубоких архитектурах, чем это было возможно ранее, без риска столкнуться с проблемой исчезающих градиентов [18].

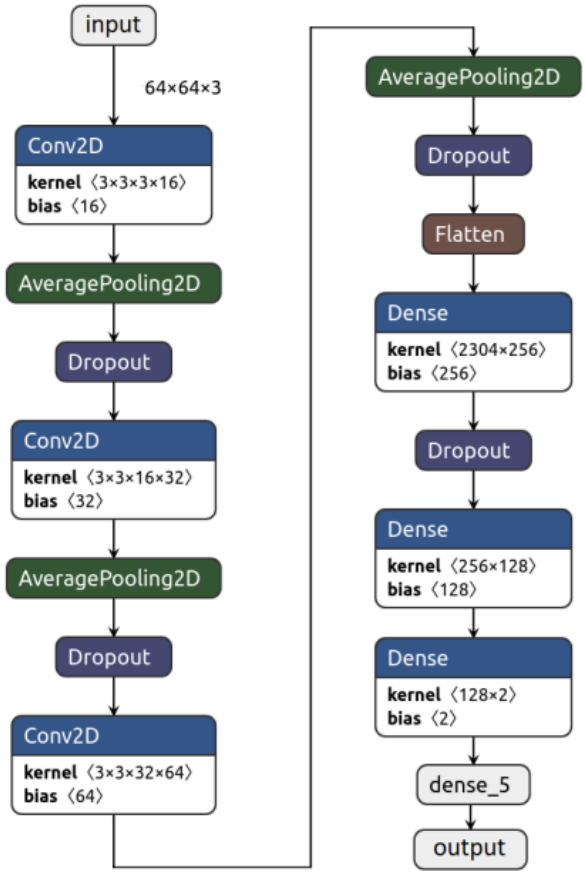


Рис. 3. Архитектура FireNet

Архитектура ResNet50 делится на четыре основные части: сверточные слои, блоки идентичности, сверточные блоки и полносвязные слои [5]. Сверточные слои отвечают за извлечение признаков из входного изображения, в то время как блок идентичности и сверточный блок отвечают за обработку и преобразование этих признаков. Наконец, полносвязные слои используются для окончательной классификации.

ResNet50 состоит из нескольких сверточных слоев, за которыми следуют пакетная нормализация и активационная функция ReLU. Эти слои отвечают за извлечение признаков из входного изображения, таких как края, текстуры и формы. За сверточными слоями следуют слои максимального пулинга, которые уменьшают пространственные размеры матриц признаков, сохраняя при этом наиболее важные признаки.

Блок идентичности и сверточный блок являются ключевыми строительными блоками ResNet50. Блок идентичности пропускает входные данные через серию сверточных слоев и добавляет входные данные обратно к выходу. Это позволяет сети обучаться по остаточным функциям, которые отображают входные данные в желаемый результат. Сверточный блок похож на блок идентичности, но с добавлением 1x1 сверточного слоя, который используется для уменьшения количества фильтров перед сверточным слоем 3x3.

Заключительной частью ResNet50 являются полносвязные слои. Эти слои отвечают за окончательную классификацию. Выход последнего полносвязного слоя подается на активационную функцию Softmax для получения окончательных вероятностей классов. Для задачи классификации изображений с огнем сеть ResNet50, которая уже была обучена на большом и разнообразном датасете ImageNet, была дообучена на тренировочном датасете FireNet [8, 10]. Дообучение включает корректировку весов последних слоев, чтобы лучше адаптировать модель к специфике задачи обнаружения огня, а также выходной слой заменяется на новый, соответствующий количеству классов (два в нашем случае – «огонь» и «нет огня»). Дообучение проводилось на тренировочном наоборе данных FireNet в течении 10 эпох.

IV. Сравнение

Было проведено сравнение сети FireNet и дообученной сети ResNet50. Сравнение проводилось на двух открытых датасетах, содержащих 871 (далее DS1) и 9553 (DS2) изображения соответственно. Для оценки качества работы классификаторов использовались следующие величины:

- TP – модель обнаружила огонь там, где оно действительно есть.
- FP – модель обнаружила огонь там, где его нет.
- FN – модель не обнаружила огонь, хотя оно присутствует на изображении.

По этим величинам были построены такие функции оценок, как:

- Точность – сколько раз модель обнаружила огонь там, где он действительно есть:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

- Полнота – сколько случаев возникновения пламени обнаружила модель от общего числа таких изображений:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

- F1-мера – гармоническое среднее между точностью и полнотой, если один из параметров стремиться к нулю, она также стремиться к нулю:

$$\text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \quad (5)$$

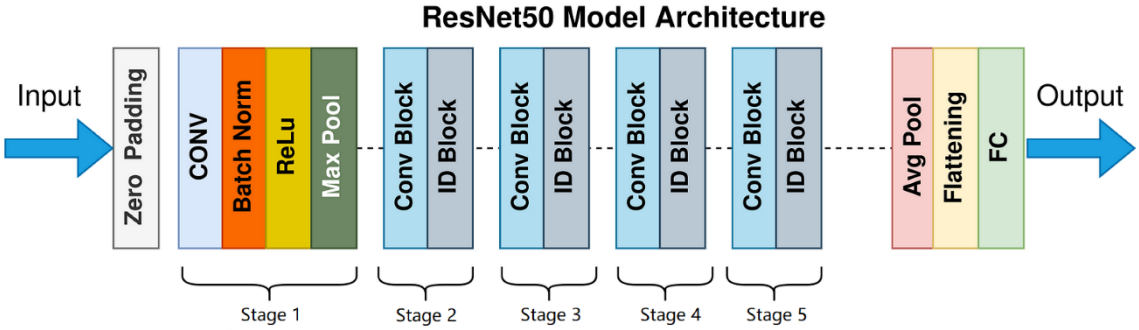


Рис. 4 Архитектура сети ResNet50

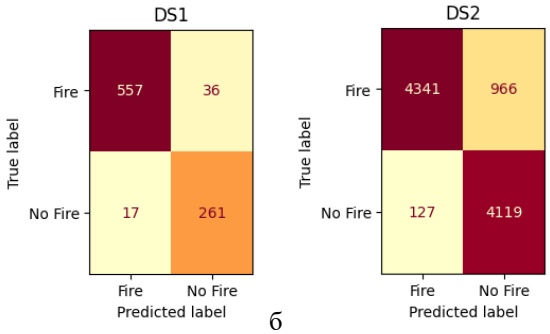
В таблице 1 показаны значения оценочных функций для двух используемых подходов. Как видно, оба подхода имеют достаточно высокие показатели исследуемых метрик, однако FireNet показывает себя лучшим классификатором из этих двух, имея лучшие показатели по всем метрикам в среднем на 21,79% что означает, что он находит больше случаев возникновения пламени и намного меньше ошибается, различая схожие с огнем по текстуре и цветовой гамме изображения.

ResNet50 склонна к большему количеству ложных классификаций, поскольку часто предсказывает отсутствие пламени на изображениях, где оно присутствует.

ТАБЛИЦА I. Сравнение метрик FireNet и ResNet50.

	FireNet		**ResNet50**	
	DS1	**DS2**	**DS1**	**DS2**
Precision	0.97	0.971	0.779	0.633
Recall	0.939	0.818	0.802	0.827
F1	0.954	0.888	0.791	0.717

Исследуемые нейронные сети на выходе имеют два класса, означающие наличие или отсутствие пламени на изображении.



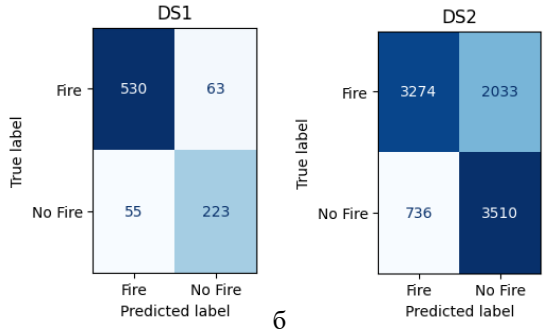
а б

Рис. 5. Матрица ошибок FireNet: а) на DS1, б) на DS2

На рисунке 5 отображена матрица ошибок на двух открытых датасетах для сети FireNet, а на рисунке 6 – для ResNet50. Из рисунка видно, что FireNet продемонстрировала хорошие показатели на обоих наборах данных, точно классифицируя изображения при различных показателях качества снимка и уровня освещения, а также успешно отличая схожие с изображениями огня кадры. Таким образом, численные оценки классификации этой нейросети имеют значения, близкие к 100%

В то же время ResNet имеет куда более скромные показатели качества, демонстрируя большое количество не идентифицированных случаев возникновения пламени, что непозволительно при практическом применении.

Такая разница в работе двух нейросетей может объяснятся особенностью их архитектур, к примеру, FireNet имеет меньшую глубину, специализированные слои и оптимизирована под конкретные признаки огня, в то время как ResNet, имеет большую глубину и может быть больше склонна к переобучению [19], также она построена для выделения более общих признаков, что оказалось не так эффективно в решении такой узконаправленной задачи, как выявление огня.



а б

Рис. 6. Матрица ошибок ResNet50: а) на DS1, б) на DS2

V. Заключение

В ходе исследования было проведено сравнение двух нейросетевых архитектур — FireNet, специально спроектированной для классификации изображений с огнем и интеграции в системы обнаружения пожара в дополнение к физическим датчикам, и ResNet50. Были рассмотрены наборы данных, используемые для обучения и тестирования, а также подробно рассмотрены архитектурные особенности обеих сетей.

Результаты сравнительного анализа показали, что FireNet имеет на 21,79% более высокие показатели качества классификации, чем ResNet50, справляясь даже с наиболее сложными изображениями, где присутствие огня трудно обнаружить. FireNet показала не только высокую точность и полноту, но и более низкое количество ложных срабатываний, что является критически важным для систем пожарной безопасности. Эти результаты доказывают, что FireNet является более качественным и специализированным решением,

превосходящим классические архитектуры, такие как ResNet, по ключевым показателям классификации в задаче обнаружения огня. Дополнительно, FireNet является более легкой и быстрой в работе, что делает её особенно полезной для реальных применений и внедрений в системы предупреждения [20, 21], где важны как скорость, так и точность обнаружения.

Литература

[1] Illarionova S, Shadrin D, Tregubova P, Ignatiev V, Efimov A, Oseledets I, Burnaev E. “A Survey of Computer Vision Techniques for Forest Characterization and Carbon Monitoring Tasks”. Remote Sensing. 2022; 14(22):5861.

[2] Savelyev B. et al. “Formalizing and securing relationships on multi-task metric learning for IoT-based smart cities” . J. Phys.: Conf. 2021, Ser. 2094

[3] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, p. 436, 2015.

[4] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, University of Oxford, conference paper at ICLR 2015.

[5] R. F Kaiming HeXiangyu ZhangShaoqing RenJian Sun. “Deep Residual Learning for Image Recognition”. Conference: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[6] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, “Convolutional neural networks based fire detection in surveillance videos,” IEEE Access, vol. 6, pp. 18174–18183, 2018.

[7] Sathishkumar, V.E., Cho, J., Subramanian, M. et al. Forest fire and smoke detection using deep learning-based learning without forgetting. fire ecol 19, 9 (2023).

[8] Jadon, Arpit and Omama, Mohd and Varshney, Akshay and Ansari, Mohammad Samar and Sharma, Rishabh. “Firenet: A specialized lightweight fire & smoke detection model for real-time iot applications”. arXiv preprint arXiv:1905.11922, 2019.

[9] “Raspberry Pi 3 Model B.” https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. Accessed: 2019-14-03.

[10] P. Foggia, A. Saggese, and M. Vento, “Real-time fire detection for videosurveillance applications using a combination of experts based on color,shape, and motion,” IEEE TRANSACTIONS on circuits and systems forvideo technology, vol. 25, no. 9, pp. 1545–1556, 2015.

[11] Deng, J. et al., 2009. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255.

[12] LTTTNT. “FIRE Dataset” Open Source Dataset, available at: https://universe.roboflow.com/ltttnt/fire-vqbia (Accessed: May 8, 2024).

[13] “Non-Fire Dataset” Open Source Dataset, available at: https://universe.roboflow.com/nonfire/non-fire-evyn3 (Accessed: May 8, 2024).

[14] Chernyshova, Y. S. ; Savelyev, B. I. ; Solodov, S. V. ; Pronichkin, S. V. “Applying distributed ledger technologies in megacities to face anthropogenic burden challenges”. IOP Conference Series: Earth and Environmental Science, 2022, Volume 1069

[15] Anokhin, K.V. , Novoselov, K.S. , Smirnov, S.K. , Efimov, A.R. , Matveev, P.M. “AI for Science and Science for AI”. Voprosy Filosofii, 2022. V.3, P.93-106

[16] Khan, Riaz Ullah & Zhang, Xiaosong & Kumar, Rajesh & Opoku Aboagye, Emelia & Kumar, Raja. (2018). Evaluating the Performance of ResNet Model Based on Image Recognition. ICCAI 2018 Proceedings of the 2018 International Conference on Computing and Artificial Intelligence

[17] Nazmul Shahadat, Anthony S. Maida. “Enhancing ResNet Image Classification Performance by using Parameterized Hypercomplex Multiplication”. University of Louisiana at Lafayette, 2018.

[18] Hochreiter, Sepp. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions”, International Journal of Uncertainty, International Journal of Uncertainty. 1998. V.6.

[19] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, Barret Zoph. “Revisiting ResNets: Improved Training and Scaling Strategies”. NeurIPS 2021 Conference, 2021.

[20] Trofimov V. B., Temkin I. O., Solodov S. V. Application of case-based reasoning in hazard evaluation in complex process flow control, Eurasion mining, 2021, V.2

[21] Ghali, Rafik & JMAL, Marwa & Mseddi, Wided & Attia, Rabah. (2020). Recent Advances in Fire Detection and Monitoring Systems: A Review. Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT’18), Vol.1

ГРНТИ 28.23.15
ASJC 1707
OECD 01.02.EP

Применение компьютерного зрения для определения пола человека по фотографии

Л. С. Измайлов
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m1900850@edu.misis.ru

А. М. Устинов
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
m1902107@edu.misis.ru

***Аннотация* — в данной статье рассматривается применение методов компьютерного зрения для определения пола человека по фотографии. Современные технологии машинного и глубокого обучения позволяют создавать эффективные алгоритмы распознавания образов, способные автоматически определять пол лица на изображении. Обзор существующих подходов к данной задаче включает в себя описание методов предобработки изображений, выбора признаков, а также применение различных архитектур нейронных сетей. В заключение представлены перспективы дальнейших исследований в этой области и потенциальные применения данной технологии в реальных сценариях.**

Ключевые слова — Компьютерное зрение, Детектирование гендера, Анализ моделей, CV, VGG19, MobileNetV2

I. Введение

Искусственные нейронные сети представляют собой современный и важный класс алгоритмов машинного обучения, вдохновленных организацией и функциональностью нейронов в человеческом мозге. Эти сети состоят из взаимосвязанных узлов, нейронов, объединенных в слои, которые могут обрабатывать информацию, выявлять закономерности и делать предсказания. Искусственные нейронные сети находят применение в различных областях, включая системы навигации [1], распознавание образов [2], обработку естественного языка [3], компьютерное зрение [4] и многих других. Например, в задачах классификации изображений глубокие нейронные сети могут автоматически распознавать и классифицировать объекты на фотографиях. В области обработки естественного языка они успешно используются для автоматического перевода текстов, распознавания речи и создания чат-ботов. Искусственные нейронные сети также применяются в задачах прогнозирования, управления процессами и в робототехнике [5], что подчеркивает их универсальность и эффективность в различных областях применения.

Компьютерное зрение – это область исследований в информатике, которая занимается разработкой методов и алгоритмов для обработки, анализа и интерпретации изображений и видео с помощью компьютеров. Основная цель компьютерного зрения состоит в том, чтобы обеспечить компьютерам способность "видеть" и понимать мир визуально, аналогично тому, как это делают люди.

В области компьютерного зрения используются методы машинного обучения, глубокого обучения и различные техники обработки изображений для решения разнообразных задач, таких как распознавание объектов, классификация изображений, сегментация, реконструкция 3D-моделей, распознавание жестов, а также многое другое.

В последние десятилетия CV – компьютерное зрение стало одним из наиболее динамично развивающихся направлений исследований в области искусственного интеллекта и компьютерной науки. Оно нашло широкое применение в самых различных сферах человеческой деятельности, начиная от медицины и биометрии и заканчивая автономными транспортными средствами и системами безопасности.

Одним из интересных и важных направлений применения компьютерного зрения является определение пола человека по фотографии. Это не только вызывает научный интерес, но и имеет практическое значение в таких областях, как системы безопасности, маркетинг и медицина. Возможность автоматического определения пола по фотографии может быть полезна для идентификации личностей, управления контентом в социальных сетях, персонализации рекламы и многих других приложений.

II. Наборы данных

Для проведения дообучения и тестирования представленных в данном исследовании нейронных сетей были задействованы конкретные наборы данных. Рассмотрим более подробно открытые датасеты, которые были использованы в ходе исследования.

A. Gender Classification Dataset

Датасет [6] представляет собой обрезанные изображения лиц мужчин и женщин. Он разделен на каталог обучения и проверки. Обучение содержит около 23000 изображений (Рисунок 1) каждого класса, а каталог проверки содержит около 5500 изображений (Рисунок 2) каждого класса. На рисунке 3 представлены примеры изображений.

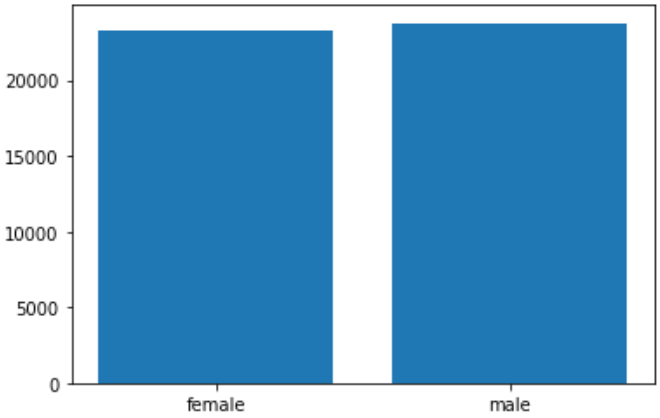


Рис. 1. Тренировочная выборка датасета

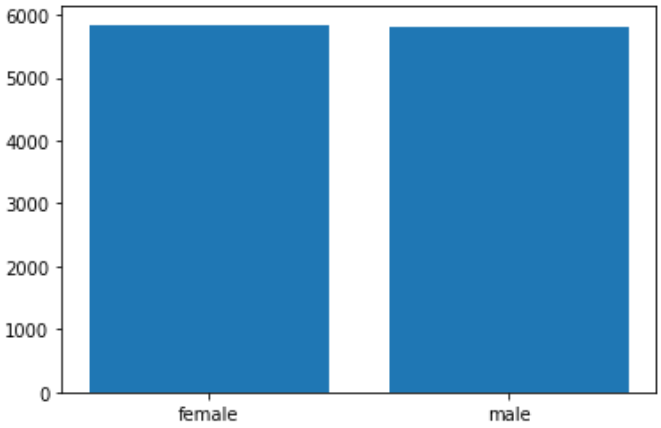


Рис. 2. Тестовая выборка датасета

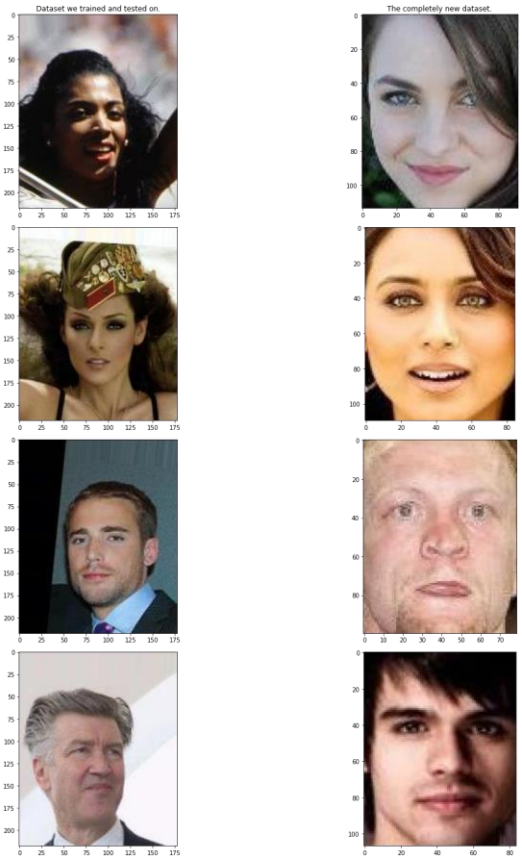


Рис. 3. Пример изображений, представленных в датасете

B. *Gender Classification 200K Images | CelebA*

Набор данных Gender Classification 200K Images | CelebA [7] используется для гендерной классификации с изображениями. Датасет состоит из почти 200 тысяч изображений размером почти 1,3 ГБ. Этот набор данных предварительно обработан из набора данных CelebFace, созданного Джессикой Ли [8]. На рисунках 4 и 5 представлены тренировочные и тестовые подвыборки, а также пример изображений.

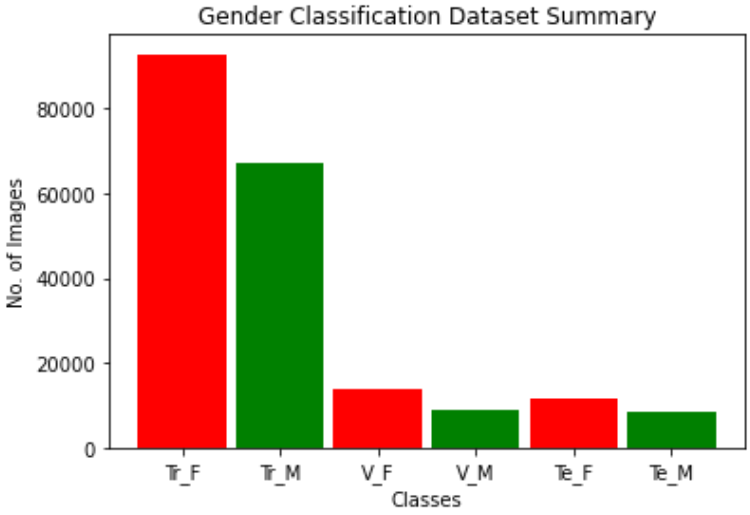


Рис. 4. Тренировочные и тестовые подвыборки датасета

Рис. 5. Пример изображений датасета

III. Подходы бинарной классификации изображений

A. *Логистическая регрессия*

Логистическая регрессия – это один из наиболее распространенных методов бинарной классификации в машинном обучении изображенная на рисунке 6. Она используется для прогнозирования вероятности отнесения объекта к одному из двух классов на основе входных признаков. Например, является ли электронное письмо спамом или не спамом, или классификация медицинского образца на наличие заболевания или его отсутствие.

Основная идея логистической регрессии – моделирование вероятности принадлежности объекта к классу с использованием логистической функции [9]. Логистическая функция, также известная как сигмоидная функция, преобразует линейную комбинацию входных признаков с их весами в вероятность принадлежности к одному из

классов. Формула (1) логистической функции выглядит следующим образом:

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

где z - линейная комбинация входных признаков с их весами, а e - математическая константа экспонента.

Веса модели настраиваются в процессе обучения на обучающем наборе данных с известными метками классов. Для настройки весов используется метод максимального правдоподобия или другие оптимизационные методы, такие как градиентный спуск [10].

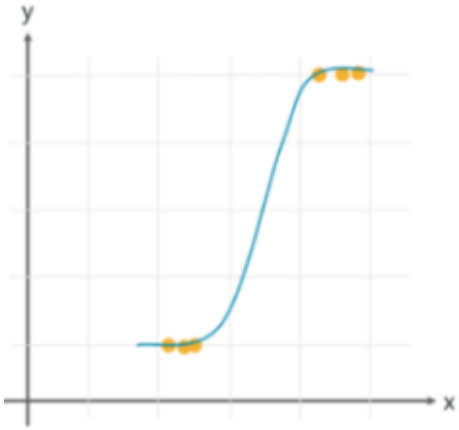


Рис. 6. Работа логистической регресии

B. *Дерево решений*

Дерево решений – это один из наиболее популярных методов машинного обучения для решения задач классификации и регрессии. Оно представляет собой древовидную структуру, состоящую из узлов (вершин) и ребер (ветвей), где каждый узел содержит условие на одном из признаков данных, а каждое ребро соответствует разделению данных на основе значения этого признака. Корневой узел дерева – это начальное условие, а листовые узлы – это конечные решения, такие как прогнозируемый класс или значение целевой переменной.

Процесс построения дерева решений начинается с выбора признака, по которому будет производиться разделение данных на наиболее чистые подмножества [11]. Для этого используются различные критерии, такие как критерий Джини или энтропийный критерий, которые измеряют неоднородность данных в узле. На рисунке 7 видно, чем меньше неоднородность, тем лучше разделение.

Одна из основных преимуществ деревьев решений – их способность автоматически выявлять нелинейные зависимости в данных, а также возможность интерпретации решений, принимаемых моделью. Деревья решений также могут быть использованы для решения задач регрессии, где вместо классов прогнозируется числовое значение целевой переменной.

Однако, деревья решений также имеют некоторые ограничения. Они могут быть склонны к переобучению, особенно если дерево слишком глубокое и сложное, что может привести к плохой обобщающей способности на новых данных. Для борьбы с переобучением можно использовать техники ограничения глубины дерева, отсечения ветвей с малым количеством объектов, а также ансамблевые методы, такие как случайный лес.

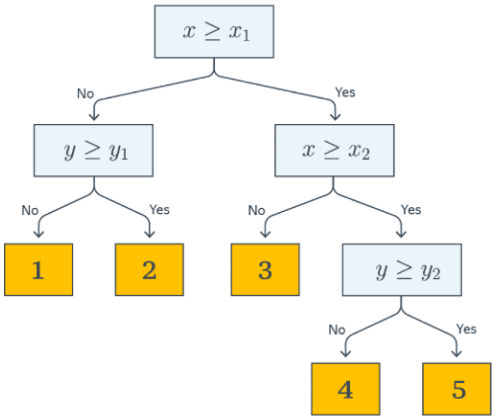


Рис. 7. Работа дерева решений

C. *SVM*

Метод опорных векторов (Support Vector Machine, SVM) – это алгоритм машинного обучения, который используется для решения задачи бинарной классификации, то есть разделения данных на два класса. Основной целью SVM является поиск оптимальной разделяющей гиперплоскости, или линии, которая наилучшим образом разделяет данные на два класса [12].

Принцип работы SVM основан на поиске опорных векторов – точек данных, которые находятся наиболее близко к разделяющей гиперплоскости на рисунке 8. Эти опорные векторы определяют положение гиперплоскости и влияют на ее определение. Алгоритм SVM стремится максимизировать расстояние между разделяющей гиперплоскостью и опорными векторами, что называется "зазором". Таким образом, лучшей гиперплоскостью считается та, для которой зазор максимально велик.

Для нахождения оптимальной гиперплоскости SVM использует различные методы оптимизации и выбора гиперплоскости, такие как методы максимального зазора, мягкой максимальной разделимости, и ядерные методы. Метод максимального зазора стремится найти гиперплоскость, которая максимизирует зазор между классами, что обеспечивает лучшую обобщающую способность модели. Мягкая максимальная разделимость позволяет допускать нарушения в разделении классов, чтобы модель была более гибкой и устойчивой к шумным данным. Ядерные методы позволяют применять SVM к нелинейно разделимым данным, проецируя их в более высокоразмерное пространство с помощью ядерных функций [13].

Одним из преимуществ SVM является его способность эффективно работать с данными, имеющими большое количество признаков, что можно увидеть на рисунке 9. Кроме того, SVM также имеет возможность работать с несбалансированными данными, где один класс может быть редким [14]. Однако, SVM также имеет некоторые ограничения, такие как чувствительность к выбросам и шуму в данных, а также сложность интерпретации результатов.

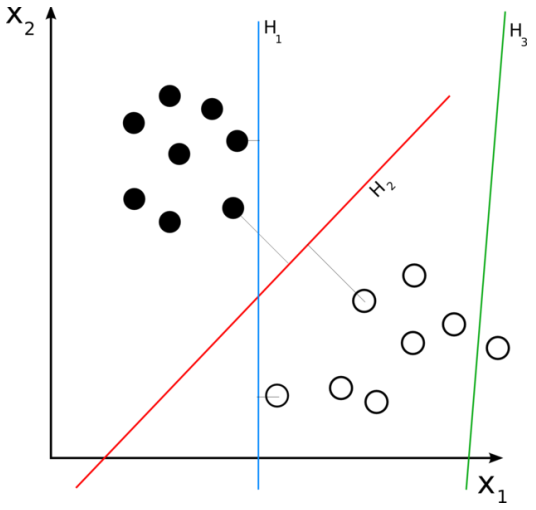


Рис. 8. Работа метода опорных точек

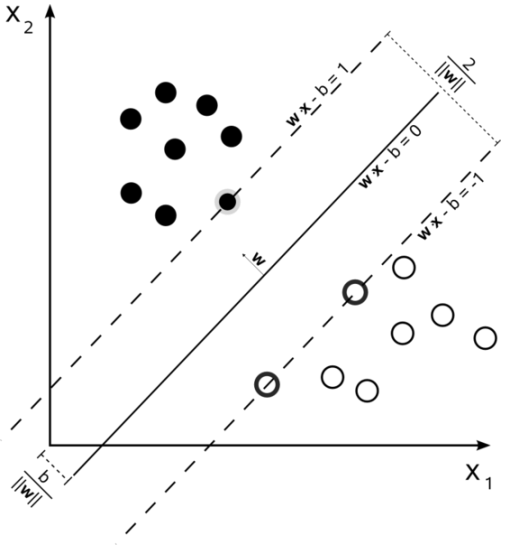


Рис. 9. Описание работы метода опорных точек

D. Алгоритм k ближайших соседей

Алгоритм k ближайших соседей представляет собой метод классификации, который состоит из трех последовательных этапов [15]:

На первом этапе (Рисунок 10), для целевого объекта, который нужно классифицировать, вычисляется расстояние до каждого объекта обучающей выборки, уже отмеченного определенным классом.

На втором этапе (Рисунок 11) выбирается k объектов обучающей выборки, расстояния до которых минимальны. Изначально значение k выбирается произвольно, однако, на следующих итерациях оно подбирается исходя из точности прогнозов, полученных для каждого выбранного значения k.

На третьем этапе (Рисунок 12) класс целевого объекта определяется на основе наиболее часто встречающегося класса среди k ближайших соседей. Класс может быть числовым или текстовым значением, в зависимости от способа обозначения классов в исходных данных. Например, в случае с беспилотными летательными аппаратами, классы могут быть обозначены как "человек" или "бетонный блок".

Конечная точность и эффективность алгоритма k ближайших соседей зависит от нескольких факторов, таких как выбор метрики расстояния, значение k и предобработка данных. Важно выбрать подходящую метрику расстояния, которая будет использоваться для определения близости между объектами. Наиболее распространенные метрики включают евклидово расстояние, манхэттенское расстояние, и расстояние Минковского, однако выбор метрики должен быть основан на спецификах задачи и характеристиках данных.

Предобработка данных также важна для успешной работы алгоритма k ближайших соседей. Некорректные или неполные данные могут привести к неправильным предсказаниям. Поэтому перед применением алгоритма необходимо произвести очистку данных, заполнение пропущенных значений, нормализацию или стандартизацию признаков и другие преобразования данных.

Преимущества алгоритма k ближайших соседей включают простоту реализации, способность работать с несбалансированными данными и способность адаптироваться к изменениям в данных [16]. Однако, у него также есть некоторые ограничения, такие как низкая скорость работы при большом объеме данных, чувствительность к шуму и неспособность моделировать сложные нелинейные отношения между признаками.

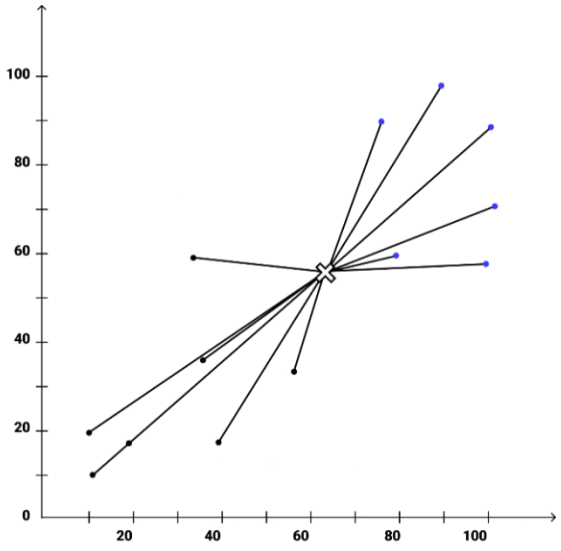


Рис. 10. Вычисление расстояние от обьекта

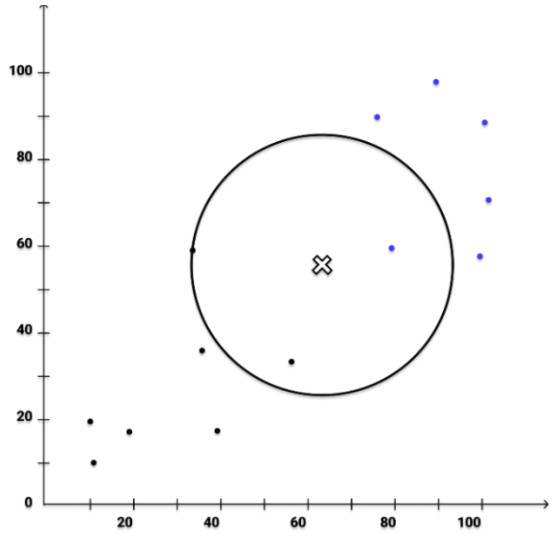


Рис. 11. Выборка объектов с мин. Расстоянием

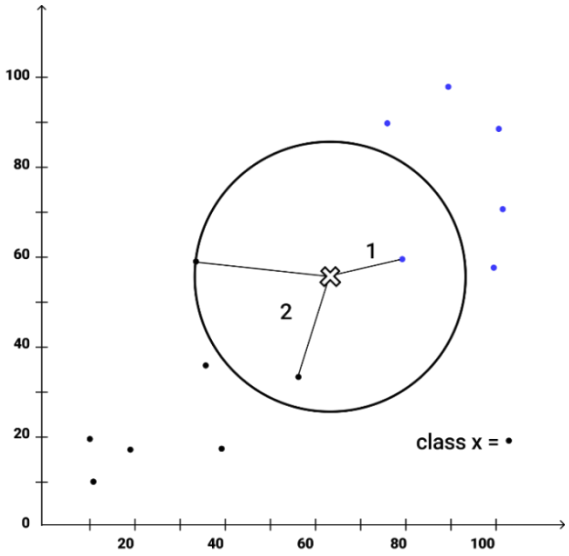


Рис. 12. Получение класса объекта

IV. Нейросетевые архитектуры

A. *VGG19*

VGG19 – это нейронная сеть глубокого обучения, которая была разработана и представлена в 2014 году в работе "Very Deep Convolutional Networks for Large-Scale Image Recognition" исследовательской группой Visual Geometry Group (VGG) при Оксфордском университете.

Нейронная сеть VGG19 представляет собой сверточную нейронную сеть, состоящую из 19 слоев, включая 16 сверточных слоев и 3 полносвязанных слоя (Рисунок 13). Она является одной из первых глубоких сверточных нейронных сетей, которая показала высокую точность в задачах классификации изображений на наборе данных ImageNet.

Особенностью архитектуры VGG19 является использование множества сверточных слоев с небольшим размером ядра (3x3) и максимальным пулингом (MaxPooling) после каждых двух сверточных слоев, что способствует извлечению более абстрактных признаков из изображений.

VGG19 стала популярной моделью в области компьютерного зрения и глубокого обучения благодаря своей простоте и хорошей обобщающей способности на различных наборах данных. Она часто используется в качестве базовой модели или предобученной модели для решения различных задач, таких как классификация изображений, детектирование объектов и сегментация изображений.

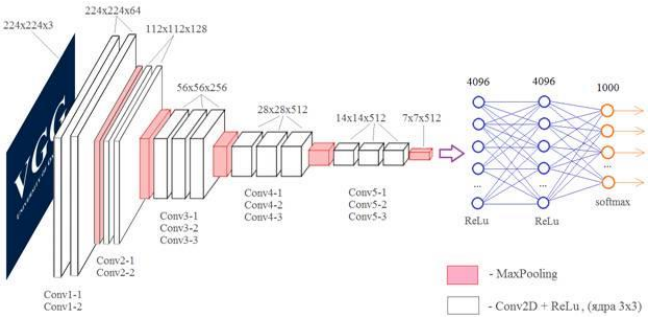


Рис. 13. Общая структура сети VGG-19

B. *MobileNetV2*

MobileNetV2 – это архитектура сверточной нейронной сети, разработанная Google в 2018 году как улучшение оригинальной архитектуры MobileNet, цель которой – создание компактных и быстрых моделей для развертывания на мобильных устройствах и встроенных системах.

MobileNetV2 была разработана с учетом необходимости улучшения эффективности и точности по сравнению с первым поколением MobileNet. Эта архитектура достигла лучшей производительности по сравнению с оригинальным MobileNet, при этом сохраняя компактность и высокую скорость работы.

Основные особенности MobileNetV2 включают в себя:

- использование блока "Inverted Residuals with Linear Bottlenecks". Данный блок состоит из последовательности операций свертки, активации, свертки с уменьшением размерности и линейной проекции, что позволяет уменьшить количество параметров модели, сохраняя при этом ее эффективность;
- использование блока "Inverted Residuals with Linear Bottlenecks". Этот блок состоит из последовательности операций свертки, активации, свертки с уменьшением размерности и линейной проекции, что позволяет уменьшить количество параметров модели, сохраняя при этом ее эффективность;
- использование слоев "Linear Bottlenecks". Они позволяют сократить количество вычислений за счет уменьшения размерности пространства признаков;
- использование метода "shortcut connections". Он позволяет улучшить процесс обучения и стабилизировать градиенты в глубоких нейронных сетях.

MobileNetV2 получила широкое распространение благодаря своей способности обеспечить высокую производительность на устройствах с ограниченными вычислительными ресурсами, таких как мобильные телефоны, встроенные системы и устройства Интернета вещей. Она успешно применяется в задачах классификации изображений, детекции объектов и сегментации изображений.

На рисунке 14 представлена общая схема сети MobileNetV2.

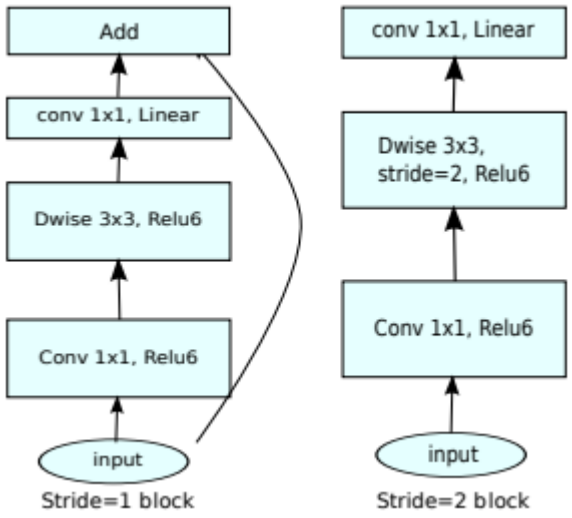


Рис. 14. Схема сети MobileNetV2

V. Оценка точности

Сравним две модели, такие как VGG19 и MobileNetV2.

Для оценки эффективности моделей мы использовали несколько метрик. Одной из наиболее распространенных является F1-мера (F1-score), которая является гармоническим средним между точностью (precision) и полнотой (recall).

TP (True Positive): количество изображений, которые были правильно классифицированы как положительные.

FP (False Positive): количество изображений, которые были ошибочно классифицированы как положительные (то есть модель сказала, что они положительные, но на самом деле они относятся к отрицательному классу).

FN (False Negative): количество изображений, которые были ошибочно классифицированы как отрицательные (то есть модель сказала, что они отрицательные, но на самом деле они относятся к положительному классу).

Recall (Полнота): отношение TP к общему числу изображений, которые действительно принадлежат к положительному классу. Полнота измеряет, насколько хорошо модель обнаруживает все положительные комментарии.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Precision (Точность): отношение TP к общему числу изображений, которые модель предсказала как положительные. Точность измеряет, насколько точными являются положительные предсказания модели.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

F1 Score: это гармоническое среднее между точностью и полнотой. F1 Score высок, если и точность, и полнота высоки. F1 Score учитывает и точность, и полноту, предостерегая от переоценки модели, которая может быть высокой по одному из этих показателей, но низкой по-другому.

$$F1 = 2 * Precision * \frac{Recall}{Precision + Recall} \quad (4)$$

В таблицах I и II представлена информация о метриках моделей на датасетах.

Исходя из F1 метрики на первом датасете, можно заметить, что VGG19 показывает себя лучше, чем модель MobileNetV2 на 0.05. На тестовой выборке мужчины угадывались лучше, чем женщины, потому что количество фотографий мужчин было больше при обучении двух моделей. Для этого датасета модели показали хорошие результаты, которые удовлетворяют решению задачи.

Для второго датасета модели показали результаты чуть хуже, чем для первого. VGG19 также сработал лучше, чем MobileNetV2, но уже на 0.13 по F1 метрике. Отметим, что разница в качестве классификации увеличилась, скорее всего это произошло из-за того, что в нем преобладают фотографии лиц азиатской внешности, что осложняет детектирование гендера человека.

На основании полученных данных моделей на основе двух датасетов можно сделать вывод, что VGG19 показывает лучшие результаты, чем MobileNetV2.

Наконец, было произведено детектирование изображений на собственных данных, которые включают в себя 18 фотографий из которых 11 мужчин и 7 женщин. Результаты представлены на рисунках 23, 24 и 25.

ТАБЛИЦА I. Оценка детектирующей части для Gender Classification Dataset

	VGG19	**MobileNetV2**
TP	4225	3989
FP	4516	4182
FN	2259	2829
Precision	0.77	0.73
Recall	0.81	0.75
F1	0.79	0.74

ТАБЛИЦА II. Оценка детектирующей части для Gender Classification 200K Images | CelebA

	VGG19	**MobileNetV2**
TP	9854	7996
FP	8996	7531
FN	6150	9473
Precision	0.66	0.53
Recall	0.91	0.76

F1	0.76	0.63

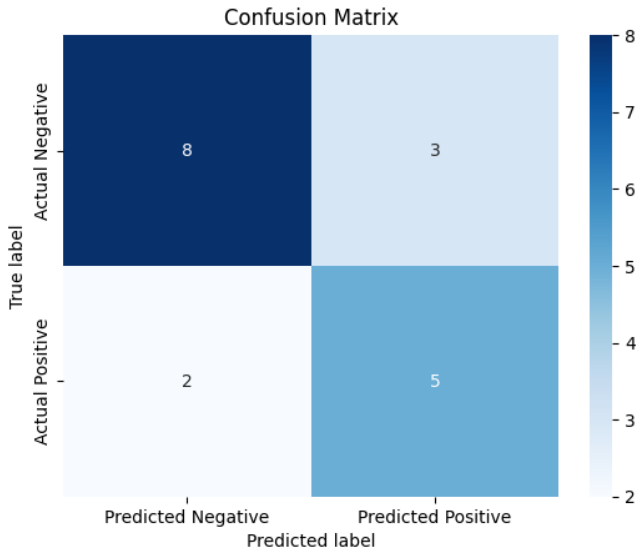


Рис. 15. Матрица точности для модели VGG19 на собственных данных

	precision	recall	f1-score	support
0	0.80	0.73	0.76	11
1	0.62	0.71	0.67	7
accuracy			0.72	18
macro avg	0.71	0.72	0.71	18
weighted avg	0.73	0.72	0.72	18

Рис. 16. Количественная оценка для модели VGG19 на собственных данных

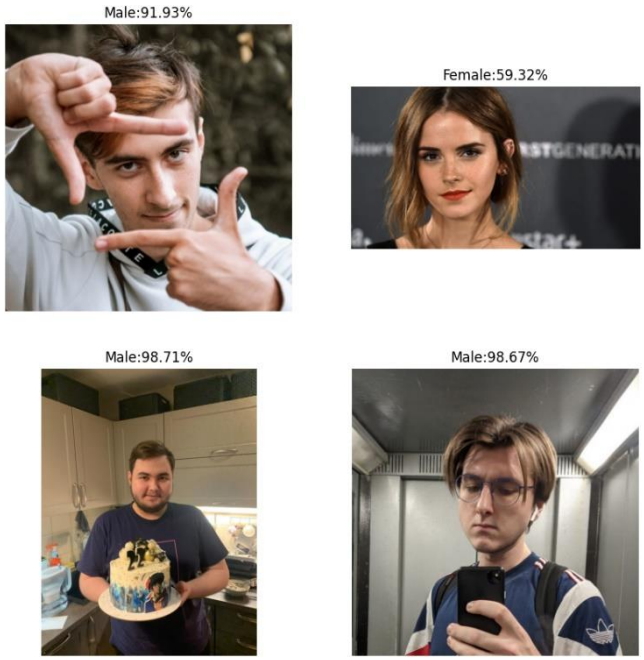


Рис. 17. Пример изображений

VI. Заключение

Были рассмотрены две модели: VGG19 и MobileNetV2. Исходя из полученных результатов при снятии метрик, стоит отметить, что для детектирования гендера человека лучше использовать модель VGG19, которая по точности в среднем превосходит MobileNetV2 на 12%. Данная модель может быть встроена в систему безопасности различных объектов или маркетинга для детектирования гендера людей. В качестве дальнейшего исследования можно рассмотреть и сравнить применение других нейронных сетей.

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] Gender Classification Dataset, available at: https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset (Accessed: May 02, 2024).

[7] Gender Classification 200K Images | CelebA, available at: https://www.kaggle.com/datasets/ashishjangra27/gender-recognition-200k-images-celeba (Accessed: May 02, 2024).

[8] CelebFaces Attributes (CelebA) Dataset, available at: https://www.kaggle.com/datasets/jessicali9530/celeba-dataset (Accessed: May 02, 2024).

[9] Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning. Data Mining, Inference and Prediction. New York.: Springer, 2017. — 446 с

[10] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin. Learning From Data. New York.: AMLbook, 2017. — 201 с.

[11] А. А. Слинкина. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных // М.: ДМК Пресс, 2016. – 400 с.

[12] Вьюгин В.В. Математические основы машинного обучения и прогнозирования. // Лаборатория структурных методов анализа данных в предсказательном моделировании МФТИ. 2018. URL: http://iitp.ru/upload/publications/8207/vyugin1.pdf (дата обращения: 02.05.2024).

[13] А. А. Слинкина. Обучение с подкреплением: Введение. 2-е изд. М.: ДМК Пресс, 2020. – 552 с.

[14] Тарик Рашид. Создаем нейронную сеть. Спб.: Альфа-книга, 2017. – 272 с.

[15] Бринк Хенрик, Ричардс Джозеф, Феверолф Марк. Машинное обучение. Спб.: Manning, 2017. – 336 с.

[16] Себастьян Рашка, Вахид Мирджалили. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2. Спб.: ООО “Диалектика”, 2020. – 848 с

ГРНТИ коды: 20.01.07, 20.53.19, 20.53.15, 28.23.15, 28.23.37, 28.23.01
ASJC коды: 1707, 1702
OECD коды: 01.02.EP

Поиск ключевых точек на изображении лица

А. А. Фомина
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2300443@edu.misis.ru

***Аннотация*—В данном научном исследовании производится анализ различных методов обнаружения ключевых точек на лице с применением нейронных сетей. Основная цель работы заключается в исследовании и выявлении наиболее эффективных моделей, способных обнаруживать и точно определять ключевые точки на лице даже в условиях сложных изображений. При этом проводится сравнительный анализ возможностей данных моделей на различных наборах данных, чтобы выявить их преимущества и недостатки в различных условиях. В ходе исследования также рассматриваются методы предварительной обработки данных, включая аугментацию, для улучшения устойчивости моделей к изменениям освещения, углов съемки и другим факторам. Полученные результаты и выводы данного исследования предоставляют важные инсайты для дальнейшего развития технологий компьютерного зрения в области распознавания лиц.**

Ключевые слова — компьютерное зрение, ключевые точки лица, аугментация данных, нейронная сеть, CNN, MLP

I. Введение

Изучение ключевых точек на лице — это актуальная и стратегически важная задача в современных исследованиях компьютерного зрения. Развитие эффективных методов обнаружения этих точек имеет огромное значение для множества областей, включая биометрию, виртуальную реальность, медицинскую диагностику и другие. Новейшие технологии в области выявления ключевых точек на лице способствуют не только улучшению систем распознавания лиц, но и повышению уровня безопасности, а также открывают новые перспективы для инноваций в сфере компьютерного зрения. Это исследование играет важную роль в развитии современных методов анализа лиц, способствуя расширению знаний в данной области.

В настоящем времени, в контексте научных разработок в области определения ключевых точек на изображениях лица, становится все более актуальной необходимость создания инновационных и точных методов идентификации лиц. В последние десятилетия методы глубокого обучения [1], включая сверточные нейронные сети (CNN), привлекли значительное внимание и показали впечатляющие результаты в различных областях, включая рассматриваемую в данной статье задачу. Эти успехи побудили меня исследовать перспективы применения подобных сетей для точного и автоматизированного обнаружения ключевых точек на лице, анализируя их визуальные особенности.

В свете этих обстоятельств, в данной работе представляется методика компьютерного зрения [2, 3] для выделения основных точек лица. В работе рассматриваются и сравниваются достижения двух нейронных сетей: многослойного персептрона и сверточной нейронной сети. Архитектура данных сетей описаны ниже в данной статье.

II. Наборы данных

В данной работе представлены два набора данных, как взятых из открытых источников, так и самостоятельно собранных. Рассмотрим используемые наборы.

A. Facial Keypoint Detection

Для исследования алгоритмов обнаружения ключевых точек на лице был использован обширный набор общедоступных данных «Facial Keypoint Detection» [4]. Данный датасет состоит из списка пикселей (упорядоченных по строкам) в виде целых чисел. Каждая строка содержит координаты (x, y) для 15 ключевых точек, которые представляют элементы лица, представленные в таблице 1.

Таблица 1. Ключевые точки лица

Английский язык	Русский язык
left_eye_center	центр левого глаза
right_eye_center	центр правого глаза
left_eye_inner_corner	внутренний угол левого глаза
left_eye_outer_corner	внешний угол левого глаза
right_eye_inner_corner	внутренний угол правого глаза
right_eye_outer_corner	внешний угол правого глаза
left_eyebrow_inner_end	внутренний конец левой брови
left_eyebrow_outer_end	внешний конец левой брови
right_eyebrow_inner_end	внутренний конец правой брови
right_eyebrow_outer_end	внешний конец правой брови
nose_tip	кончик носа
mouth_left_corner	левый угол рта
mouth_right_corner	правый угол рта
mouth_center_top_lip	центр верхней губы

mouth_center_bottom_lip	центр нижней губы

Для дальнейшей работы пиксели будут преобразованы в изображения, которые имеют размер 96x96 пикселей. Количество тренировочных изображений составит 7049. Тестовых 1783. Следует отметить, что тренировочные изображения размеченные, то есть на изображениях уже отмечены ключевые очки лица.

Также, следует отметить, что датасет охватывает максимально разных людей, с разным полом, эмоциями, аксессуарами и разным положением лица.

Рис. 1. Набор данных «Facial Keypoint Detection» а), б) различные эмоции в), г) очки различной формы, д), е) положение, ж) ребенок, з) лицо с бородой и шляпой

B. Дополненный набор данных

Данный набор данных является дополненным, по отношению к набору данных «Facial Keypoint Detection». Дополненный более 20 различными изображениями лиц людей, найденных в открытых источниках. Набор данных «Facial Keypoint Detection» сам по себе является всеобъемлющем набором. Дополнительные изображения необходимы для повышения качества проверки нейронных сетей. Также, как и в первоначальном наборе, было принято решение дополнить данные не только стандартными изображениями лиц людей, но и разнообразными изображениями, учитывающими различные эмоции, пол и национальность. Примеры дополнительных изображений отражены на рисунке 2.

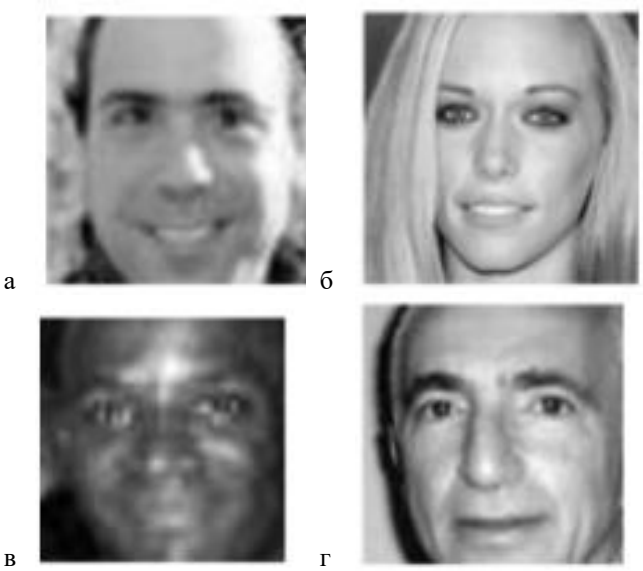


Рис. 2. Дополнительный набор данных, а), б) разный пол людей, в), г) разная национальность

III. НЕЙРОННЫЕ СЕТИ

В представленной работе для решения задачи поиска ключевых точек лица использовались две модели нейронных сетей: многослойным персептрон и сверточная нейронная сеть [5, 6]. Обе модели представляют собой мощные инструменты для извлечения высокоуровневых признаков [7] из входных данных, будь то изображения или другие структурированные данные. Это важно для точного обнаружения и определения ключевых точек на лице.

Полносвязные нейронные сети способны эффективно обрабатывать и анализировать различные типы входных данных, включая векторы, таблицы и одномерные последовательности. Они могут выявлять скрытые зависимости и закономерности в данных, что важно для решения широкого круга задач.

С другой стороны, сверточные нейронные сети (CNN) демонстрируют особую эффективность в обработке пространственных данных, таких как изображения. Благодаря операции свертки и слоям подвыборки, CNN способны извлекать иерархические пространственные признаки, что делает их оптимальным выбором для задач компьютерного зрения и распознавания образов.

Таким образом, как полносвязные, так и сверточные нейронные сети являются мощными инструментами для извлечения высокоуровневых признаков из входных данных и решения широкого спектра задач, включая точную классификацию и распознавание сложных объектов.

A. Предобработка данных

Исследуя датасет было выявлено, что у в 28 изображениях отсутствуют значения. Поскольку имеется большое количество пропущенных значений, их удаление, очевидно, уменьшит объем наших данных и затруднит наши прогнозы. В связи с этим было принято решение доразметить данные изображения. Примеры подобных изображений до и после дополнения ключевыми точками отображены ниже:

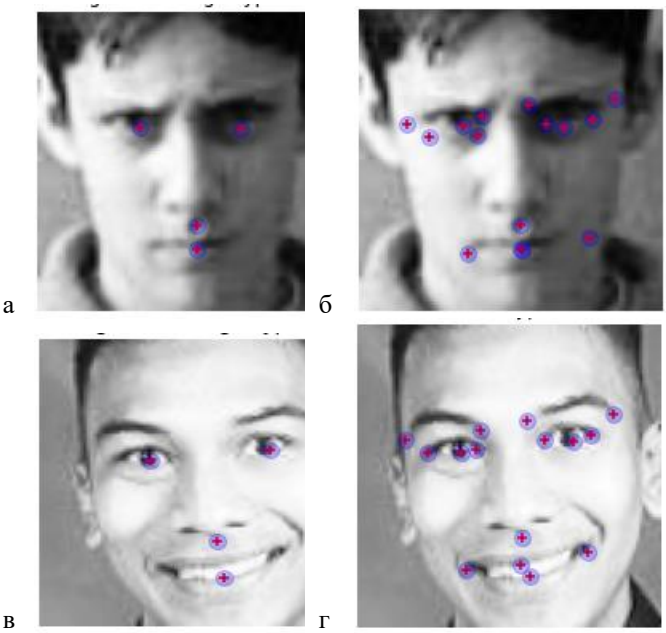


Рис. 3. Дополнительный набор данных а), в) оригинальные изображения с пропуском ключевых точек на лице, б), г) дополненные изображения

После заполнения пропущенных значений процесс обработки изображений был завершен успешно, и данные теперь соответствуют ожидаемому количеству, то есть 15 ключевым точкам на каждом изображении.

Для улучшения качества и устойчивости модели в задаче поиска ключевых точек на изображении лица была применена аугментация данных. Аугментация — это метод, который используется для преобразования исходных изображений путем применения различных трансформаций. Этот подход позволяет модели лучше обобщать и справляться с различными условиями реального мира, такими как изменение освещения, углы съемки, масштабирование и другие искажения. Примеры изображений после аугментации представлены на рисунке 4.

Рис. 4. Примеры изображений после аугментации

Основные методы аугментации, использованные в нашем исследовании, включают:

- Сдвиги и повороты: Применение случайных сдвигов и поворотов изображений позволяет модели быть устойчивой к изменению положения лица в кадре.
- Масштабирование и обрезка: Случайное увеличение или уменьшение размера изображений, а также их обрезка, помогает модели адаптироваться к изменениям расстояния до камеры.
- Изменение яркости и контраста: Варьирование яркости и контраста изображений помогает модели справляться с различными условиями освещения.
- Отражения и вращения: Применение горизонтальных и вертикальных отражений, а также небольших вращений изображений делает модель более устойчивой к различным ориентациям лица.
- Шум и размытость: Добавление случайного шума и эффекта размытия на изображения позволяет модели научиться игнорировать помехи и фокусироваться на важных признаках лица.

Аугментация данных значительно расширила наш обучающий набор, что позволило избежать переобучения модели и повысить её обобщающую способность. Более того, такие трансформации позволили модели быть более устойчивой к различным изменениям условий съемки, которые могут возникнуть в реальных приложениях, таких как системы распознавания лиц в условиях изменяющегося освещения или различных углов обзора.

B. Многослойным персептрон (MLP)

1. Архитектура модели

Модель нейронной сети представляет собой последовательную (Sequential) архитектуру с пятью слоями. Схема архитектуры нейронной сети отображена на рисунке 5.

1. Входной слой (Input Layer):

- Flatten: преобразует входные изображения лиц (размерность 96x96x1) в одномерный массив.

2. Скрытые слои (Hidden Layers):

- Dense(128, activation='relu'): Первый скрытый полносвязный слой с 128 нейронами и функцией активации ReLU.
- Dropout(0.5): Операция Dropout для предотвращения переобучения модели после первого скрытого слоя.
- Dense(128, activation='relu'): Второй скрытый полносвязный слой также с 128 нейронами и функцией активации ReLU.
- Dropout(0.5): Операция Dropout для предотвращения переобучения модели после второго скрытого слоя.
- Dense(64, activation='relu'): Третий скрытый полносвязный слой с 64 нейронами и функцией активации ReLU.

3. Выходной слой (Output Layer):

- Dense(30): Выходной слой с 30 нейронами, представляющими 15 пар ключевых точек лица (x, y координаты) для каждого лица.

2. Оптимизатор и функция потерь:

Модель компилируется с оптимизатором Adam и функцией потерь mean squared error, что указывает на то, что целью является минимизация среднеквадратичной ошибки между предсказанными и истинными значениями.

C. Сверточная нейронная сеть (CNN)

1. Архитектура модели

Эта нейронная сеть представляет собой модель глубокого обучения с последовательным (Sequential) расположением слоев. Схема архитектуры нейронной сети отображена на рисунке 6.

Она состоит из следующих компонентов:

1. Входной слой:

- Convolution2D: 32 фильтра размером 3x3, с использованием нелинейности LeakyReLU и нормализации пакетов.
- MaxPool2D: слой максимального объединения (пулинга) размером 2x2 для уменьшения размерности.

2. Скрытые сверточные слои:

Несколько слоев Convolution2D с различным количеством фильтров (от 32 до 512) и применением функции активации LeakyReLU, а также нормализации пакетов после каждого сверточного слоя.

Между некоторыми сверточными слоями также используется слой максимального пулинга для уменьшения размерности данных.

3. Полносвязные слои:

После последнего сверточного слоя модель содержит несколько полносвязных (Dense) слоев для финального преобразования выходных данных.

Используется слой Flatten для преобразования трехмерных данных в одномерный массив перед подачей на полносвязные слои.

Финальный выходной слой с 30 нейронами, представляющими 15 пар ключевых точек лица (x, y координаты) для каждого лица.

2. Оптимизатор и функция потерь:

Модель компилируется с оптимизатором Adam и функцией потерь mean squared error, что указывает на то, что целью является минимизация среднеквадратичной ошибки между предсказанными и истинными значениями.

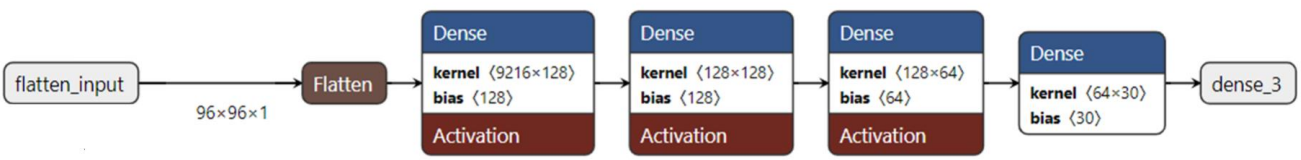


Рис. 5. Схема архитектуры MLP модели

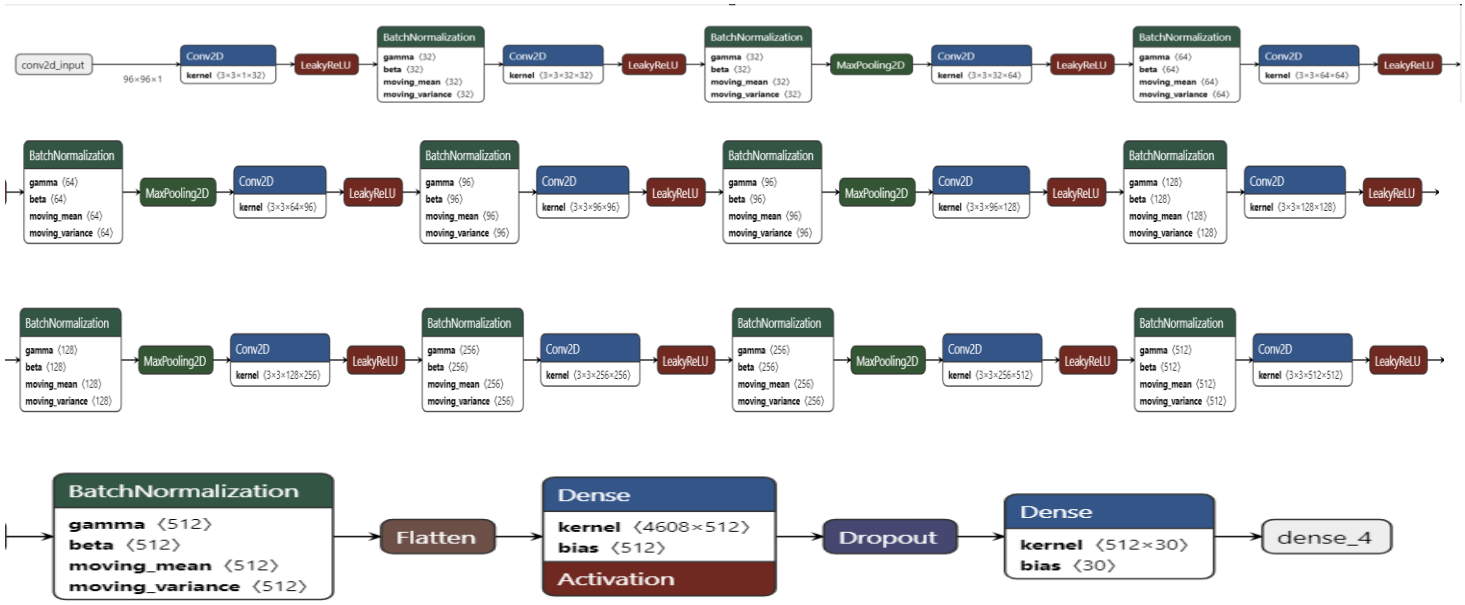


Рис. 6. Схема архитектуры CNN модели

IV. Результаты

Для оценки производительности обеих нейронных сетей в данном исследовании была использована метрика root mean squared error (RMSE). RMSE рассчитывается как квадратный корень из среднеквадратичной ошибки (MSE). Формула для вычисления RMSE выглядит следующим образом:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

где y_i — истинное значение, а y^_i — предсказанное значение, n — количество наблюдений.

A. *Многослойным персептрон (MLP)*

1. Процесс обучения:

Модель обучается на обучающих данных в течение 200 эпох с размером пакета (batch_size) равным 32. Время обучения составило 12 минут. График обучения модели отображен на рисунке 7.

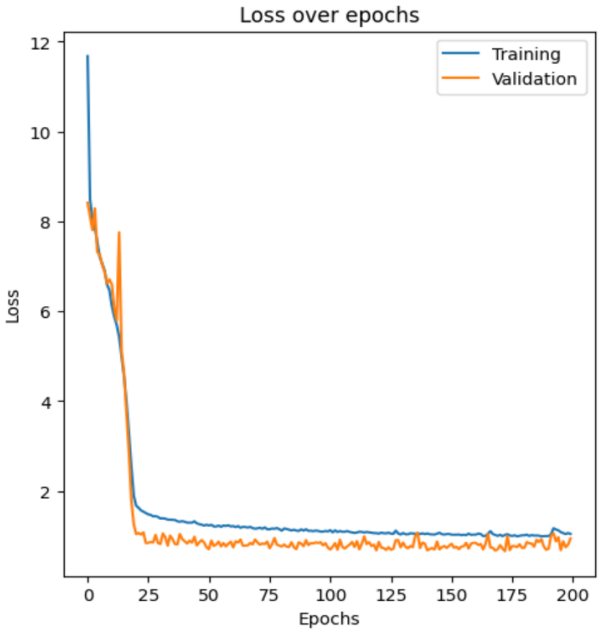


Рис. 7. График обучения MLP модели

2. Результаты обучения

По результатам обучения на протяжении 200 эпох модель продемонстрировала следующие показатели:

- Loss на тренировочной выборке: 4.8558
- RMSE на тренировочной выборке: 2.2036
- Loss на валидационной выборке: 5.5057
- RMSE на валидационной выборке: 2.3464

Эта модель представляет собой более простую архитектуру по сравнению с другой используемой моделью, с меньшим количеством слоев и без использования сверточных слоев для извлечения признаков из изображений.

B. Сверточная нейронная сеть (CNN)

1. Процесс обучения:

Модель обучается на обучающих данных в течение 50 эпох с размером пакета (batch_size) равным 256. Время обучения составило 4 часа. График обучения модели отображен на рисунке 8.

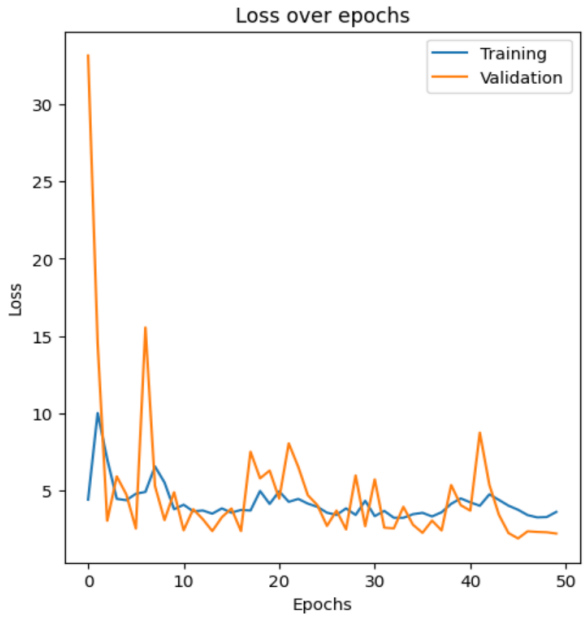


Рис. 8. График обучения CNN модели

2. Результаты обучения

По результатам обучения на протяжении 50 эпох модель продемонстрировала следующие показатели:

- Loss на тренировочной выборке: 3.5860
- RMSE на тренировочной выборке: 1.8937
- Loss на валидационной выборке: 2.1826
- RMSE на валидационной выборке: 1.4773

RMSE на валидационной выборке для каждой модели отображена в таблице:

Таблица 2. RMSE каждой модели

Модель	**RMSE**
MLP	2.1163
CNN	1.4773

Из таблицы видно, что лучшие результаты достигла CNN модель, демонстрируя RMSE в значении 1.4773. Этот показатель указывает на высокую точность предсказаний модели. Точность модели MLP составила 2.1163, что означает, что модель допускает более высокую среднеквадратичную ошибку по сравнению с моделью CNN. Это говорит о том, что модель CNN продемонстрировала более эффективную работу и показала результаты по сравнению с моделью MLP.

На рисунке отображены предсказанные моделью ключевые точки на изображениях лиц. Как видно из рисунка, модель успешно распознает и определяет основные черты лица, такие как глаза, нос и рот и другие. Визуализация демонстрирует высокую точность модели, что подтверждается низким значением среднеквадратичного отклонения (RMSE).

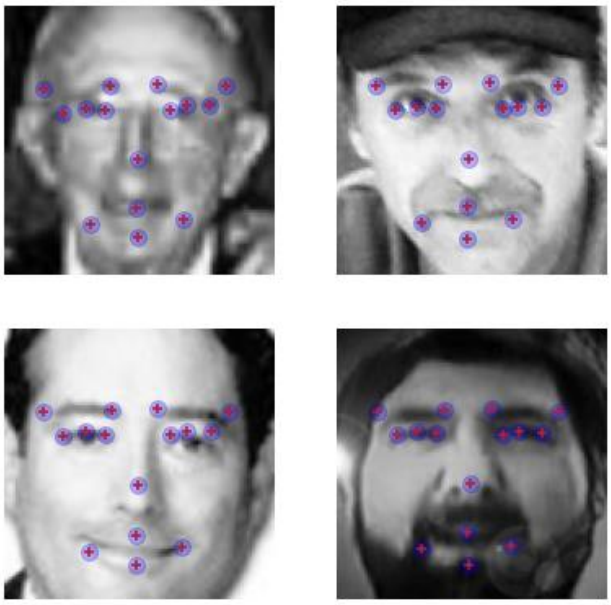


Рис. 9. Предсказанные ключевые точки лица.

I. Заключение

В данном исследовании были детально проанализированы основные наборы данных, на которых проводилось обучение и тестирование рассматриваемых нейронных сетей. Дополнительно был создан собственный набор данных для более глубокого и точного поиска ключевых точек лица, что способствовало повышению эффективности обработки информации.

Исследование включает две различные нейронные сети, используемые для решения задачи поиска ключевых точек на лице. Каждая нейронная сеть была рассмотрена с точки зрения её архитектуры и процесса обучения, что позволяет читателю получить полное представление о методологии и технических аспектах проведенного исследования.

Представленные нейронные сети были тщательно проанализированы, а результаты подвергнуты сравнительному анализу. По результатам анализа можно заключить, что модель CNN имеет определенные преимущества по сравнению с моделью MLP. Это проявляется в более высокой точности решения задачи, что

подтверждается соответствующими метриками: RMSE для CNN составил 1.4773, тогда как для MLP он был равен 2.1163.

В целом, результаты исследования подчеркивают не только значимость использования современных нейронных сетей для поиска ключевых точек на лице, но и важность выбора оптимальной архитектуры для конкретной задачи.

Литература

[1] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-Cost Navigation System for UAV," 2023 30th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), Saint Petersburg, Russian Federation, 2023, pp. 1-6, doi: 10.23919/ICINS51816.2023.10168469.

[2] 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.

[3] 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.

[4] Facial Keypoints Detection dataset, available at: https://www.kaggle.com/competitions/facial-keypoints-detection/overview

(Accessed: October 05, 2024).

[5] 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.

[6] 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.

[7] Efimoff, Albert & Matveev, Philip. (2022). Искусственный интеллект для науки и наука для искусственного интеллекта. Voprosy filosofii / Akademiia nauk SSSR, Institut filosofii. 95.10.21146/0042-8744-2022-3-93-105.

[8] Sutskever I. On the importance of initialization and momentum in deep learning / I. Sutskever, J. Martens, G. Dahl, G. Hinton // Journal of Machine Learning Research. – 2013. – V. 28, No. 3. – P. 1139–1147

[9] VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data – O'Reilly Media, 2016. – 672 p.

Классификация эмоций на лице человека при помощи компьютерного зрения

П. Д. Хонер
кафедра инженерной кибернетики
НИТУ «МИСИС»
Москва, Россия
khonerworki@gmail.com

Аннотация – в настоящее время определение эмоций человека для предоставления ему более персонифицированного опыта или для обнаружения подозрительного поведения в толпе становится все более актуальной задачей в современном обществе. В данной статье представлены различные решения задачи классификации 8 эмоций человека с использованием нейронных сетей на датасете AffectNet. Целью данного исследования является выявление наиболее эффективных моделей способных классифицировать эмоции на лице с точность выше 50%. В процессе проведения исследования сравнивается качество классификации нейронных сетей на подготовленном заранее для оценки точности небольшом наборе данных. Анализ проведенных исследований не только представляет обзор различных методов распознавания эмоций на лицах, но и подчеркивает сложность в поиске эффективных решений в этой области.

Ключевые слова — классификация эмоций, компьютерное зрение, нейронная сеть, CNN, Alex net, XGBoost, Random forest, KNN, глубокое обучение, обучение с учителем.

I. Введение

В последние десятилетия компьютерное зрение стало ключевой областью исследований в информационных технологиях, привлекающей широкий интерес в различных сферах человеческой деятельности таких как БПЛА [1] [2], автономные трамваи [3] [4] и другие. Одним из важных направлений в этой области является классификация эмоций на лицах людей при помощи компьютерного зрения. Этот подход представляет собой процесс автоматического распознавания и классификации эмоций, проявляющихся через выражения лица, с использованием различных алгоритмов и методов анализа изображений.

Исследования в области распознавания эмоций на лицах людей активно ведутся и полезны во многих областях таких как: маркетинг для создания более привлекательных рекламных компаний [5], безопасность для обнаружения подозрительного поведения [6], медицина для выявления заболеваний [7], бизнес для оценки настроения сотрудника [8] и другие.

Несмотря на свою важность и потенциал, классификация эмоций на лицах при помощи компьютерного зрения остается вызовом из-за сложности анализа человеческих эмоций и разнообразия контекстов, в которых они проявляются. Тем не менее, с развитием технологий глубокого обучения и расширением наборов данных для обучения моделей, этот подход становится все более точным и эффективным.

II. Набор данных

Для оценки работы и сравнения архитектур сетей был выбран набор данных AffectNet [9].

AffectNet – это крупнейший набор данных в мире, который используется для обучения и оценки моделей машинного обучения в области распознавания эмоций на лицах людей. Этот датасет содержит более 1 миллиона изображений лиц с разными выражениями эмоций. Изображения в AffectNet размечены семью различными эмоциональными классами и одним классом без лица: злость, смущение, отвращение, страх, счастье, нейтральность, грусть и удивление. Пример изображений с метками классов изображен на рисунке 1.

Рисунок 1 – Пример изображения из датасета AffectNet с метками классов

В данном исследовании использовалась небольшая часть данного датасета взятая с Kaggle “Facial Expressions Training Data” [10] состоящая из более чем 29 тысяч изображений разрешения 96 x 96 пикселей для экономии дискового пространства и без класса “без лица”. Пример изображений из данного датасета продемонстрирован на рисунке 2, а распределение изображений по классам продемонстрировано на рисунке 3.

Рисунок 2 – Пример изображений из датасета Facial Expressions Training Data с метками классов

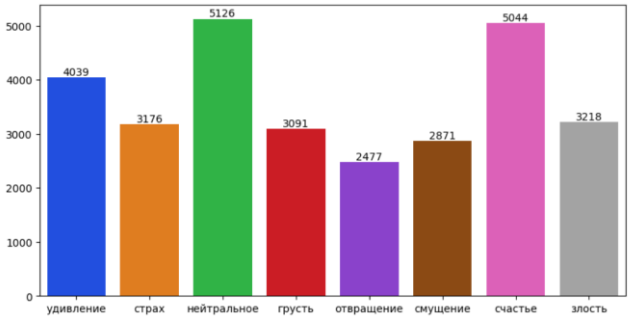


Рисунок 3 – Распределение изображений по классам

Тестирование нейронных сетей будет происходить на подготовленном заранее датасете пример изображений из него продемонстрирован на рисунке 4, а распределение изображений на рисунке 5

Рисунок 4 – Пример изображений из датасета для проверки CNN

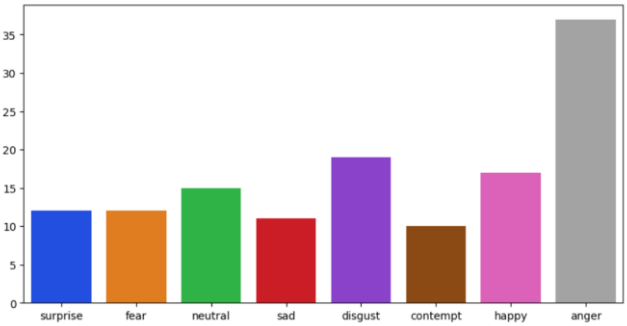


Рисунок 5 – Распределение изображений по классам

III. Нейросетевые архитектуры

A. *XGBoost*

Extreme Gradient Boosting - это библиотека для решения задач классификации, регрессии и ранжирования. Она основана на технологии градиентного бустинга решающих деревьев, ее архитектура продемонстрирована на рисунке 6.

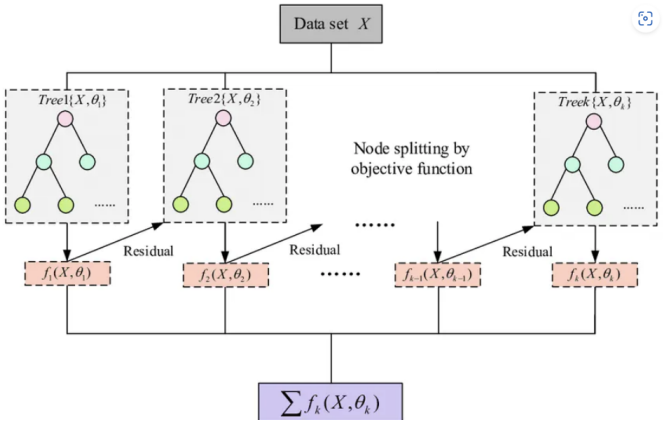


Рисунок 6 – Архитектура XGBoost

Данный алгоритм строит несколько деревьев последовательно, каждое из которых исправляет ошибки предыдущего дерева. Так же стоит сказать, что XGBoost поддерживает различные методы регуляризации, позволяющие предотвратить переобучение модели (L1 и L2). Так же данная библиотека предоставляет встроенную кросс-валидацию для оценки производительности и выбора оптимальных гиперпараметров. И наконец, XGBoost использует параллельное обучение, что позволяет значительно ускорить процесс обучения на многоядерных процессорах.

B. *KNN*

Метод k-ближайших соседей (KNN) – это алгоритм машинного обучения, который используется для решения задач классификации и регрессии. Основная идея данного алгоритма заключается в том, чтобы классифицировать новые точки данных, основываясь на классах ближайших соседей из обучающего набора данных. Архитектура данного алгоритма изображена на рисунке 7.

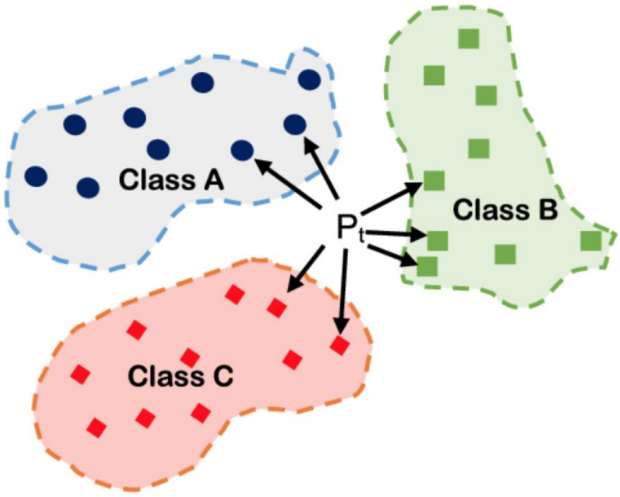


Рисунок 7 – Схема KNN

Данный алгоритм не чувствителен к выбросам, но очень чувствителен к масштабу данных. Так же плюсом является его простота. Самая большая сложность состоит в правильном выборе k (число соседей) в нашей задаче использовалось k = 5.

C. *Random Forest*

Случайный лес – это алгоритм машинного обучения, который используется для задач классификации, регрессии, выявления выбросов, кластеризации, ранжирования, прогнозирования временных рядов и других. Данный алгоритм основан на идеи использования ансамбля деревьев решений, где несколько решений деревьев обучаются на различных под выборках обучающего набора данных, а затем их результаты объединяются для получения более точного класса предсказаний. Схема Random Forest продемонстрирована на рисунке 8.

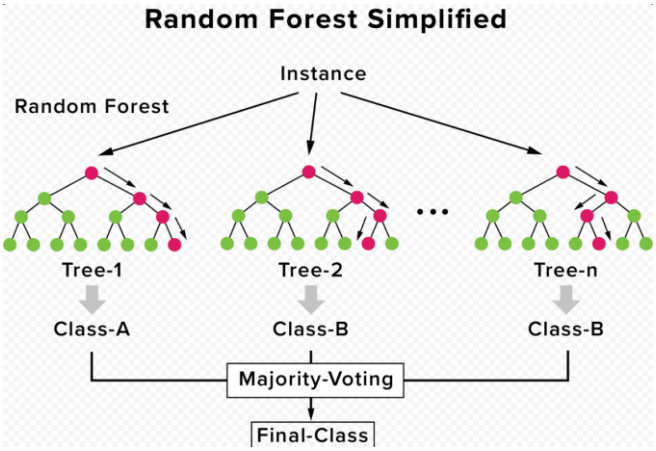


Рисунок 8 – Схема Random Forest

Данный алгоритм хорошо масштабируется и может эффективно обрабатывать большие объемы данных, а также имеет устойчивость к переобучению по сравнению с решениями, базирующимися на отдельных деревьях решений. Случайный лес достаточно универсальный алгоритм, который не требует сложной настройки гиперпараметров.

D. *CNN*

Сверточная нейронная сеть – это глубокие нейронные сети, которые получили широкое распространение в задачах обработки изображений и видео. Основное их отличие от традиционных нейронных сетей заключается в том, что они автоматически и эффективно могут выявлять важные признаки из изображений, не требуя ручного извлечения характеристик. Это делает CNN идеальными для таких задач, как распознавание лиц, автоматическое управление транспортом, медицинская диагностика и многих других.

В нашей задаче использовалась CNN состоящая из 7 слоев. 3 – сверточных слоя, 2 – слоя пулинга и 2 полносвязных слоя. В качестве функции активации использовалась relu, а на последнем слое – softmax. Оптимизатор Adam. Схема классической CNN приведена на рисунке 9.

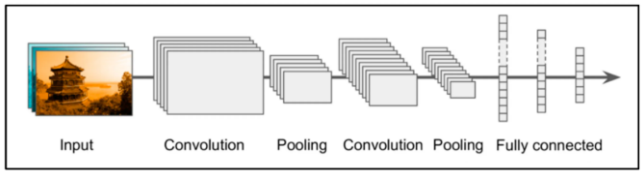


Рисунок 9 – Схема CNN

E. *Alex net*

AlexNet – это архитектура нейронной сети, которая была разработана для задач классификации изображений и последующего выявления важных признаков в данных. AlexNet, описанная в работах по глубокому обучению, использует слои свертки для извлечения важных признаков из изображений. Ее архитектура изображена на рисунке 10. Alex Net содержит в себя 5 сверточных слоев и 3 полно связных

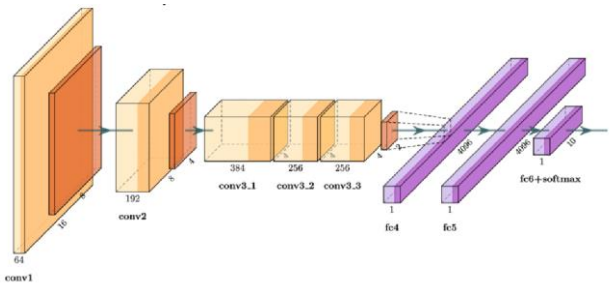


Рисунок 10 – Архитектура Alex Net

IV. Сравнение

В каждой модели использовалось разбиение на выборки Train (обучающей) состоящей из 16,414 наблюдений и 9,216 признаков и Test (проверочных) состоящей из 4,104 наблюдений и 9,216 признаков.

В качестве основных метрик для проверки использовались: Accuracy, Precision, Recall, F1. Давайте рассмотрим каждую из них подробнее:

Точность (Accuracy) – это метрика, используемая для оценки качества модели классификации. Она измеряет долю правильный предсказаний модели относительно общего количества примеров и считается по формуле (1):

$$\text{Accuracy} = \frac{\text{Количество правильных предсказаний}}{\text{Общее количесвто примеров}} \quad (1)$$

Точность (Precision) – это метрика измеряет насколько много из всех положительных предсказаний на самом деле являются верными и считается по формуле (2):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Где TP (true positives) - количество истинных положительных предсказаний, FP (false positives) - количество ложноположительных предсказаний.

Полнота (Recall) – это метрика измеряет насколько много истинных положительных результатов (TP) смогла найти модель из все возможных истинных положительных и считается по формуле (3):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

Где TP (true positives) - количество истинных положительных предсказаний, а FN (false negatives) - количество ложноотрицательных предсказаний.

F1-score – это метрика для расчета гармонического среднего между precision и recall. Рассчитывается по формуле (4):

$$\text{F1} = 2 \times \frac{Presion \times Recall}{Presion + Recall} \quad (4)$$

Перейдем к результатам. Для модели XGBoost на тестовой выборке получились результаты, изображенные на рисунке 11, а для обучающей на рисунке 12. Матрица предсказаний изображена на рисунке 13.

XGBoost model classfication report

	precision	recall	f1-score	support
surprise	0.60	0.62	0.61	780
neutral	0.81	0.88	0.84	1046
sad	0.56	0.48	0.52	625
happy	0.90	0.86	0.88	1033
anger	0.61	0.61	0.61	620
accuracy			0.72	4104
macro avg	0.69	0.69	0.69	4104
weighted avg	0.72	0.72	0.72	4104

Рисунок 11 – Результаты XGBoost на Test

XGBoost model classfication report

	precision	recall	f1-score	support
surprise	1.00	1.00	1.00	3259
neutral	1.00	1.00	1.00	4080
sad	1.00	1.00	1.00	2466
happy	1.00	1.00	1.00	4011
anger	1.00	1.00	1.00	2598
accuracy			1.00	16414
macro avg	1.00	1.00	1.00	16414
weighted avg	1.00	1.00	1.00	16414

Рисунок 12 – Результаты XGBoost на Train

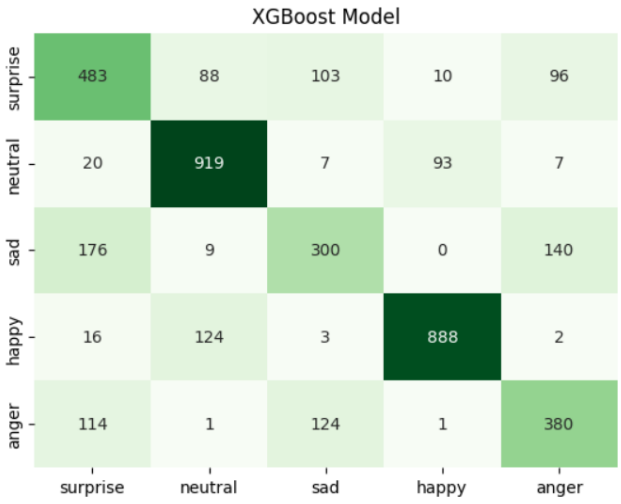


Рисунок 13 – Матрица предсказаний XGboost

Для данной модели оказалось тяжелее всего предсказывать эмоцию грусти, так мы можем увидеть явное переобучение модели на тренировочной выборке. Но в целом результат получился неплохим, все эмоции прошли порог случайного угадывания и наиболее простыми эмоциями оказались радость и нейтральность коих было больше всего в датасете.

Для модели KNN на тестовой выборке получились результаты, изображенные на рисунке 14, а для обучающей на рисунке 15. Матрица предсказаний изображена на рисунке 16.

KNN model classfication report

	precision	recall	f1-score	support
surprise	0.38	0.32	0.35	780
neutral	0.44	0.50	0.47	1046
sad	0.34	0.36	0.35	625
happy	0.46	0.63	0.53	1033
anger	0.46	0.13	0.20	620
accuracy			0.42	4104
macro avg	0.42	0.39	0.38	4104
weighted avg	0.42	0.42	0.40	4104

Рисунок 14 – Результаты KNN на Test

KNN model classfication report

	precision	recall	f1-score	support
surprise	0.43	0.35	0.39	3259
neutral	0.47	0.55	0.51	4080
sad	0.38	0.42	0.40	2466
happy	0.50	0.67	0.57	4011
anger	0.58	0.20	0.29	2598
accuracy			0.46	16414
macro avg	0.47	0.44	0.43	16414
weighted avg	0.47	0.46	0.45	16414

Рисунок 15 – Результаты KNN на Train

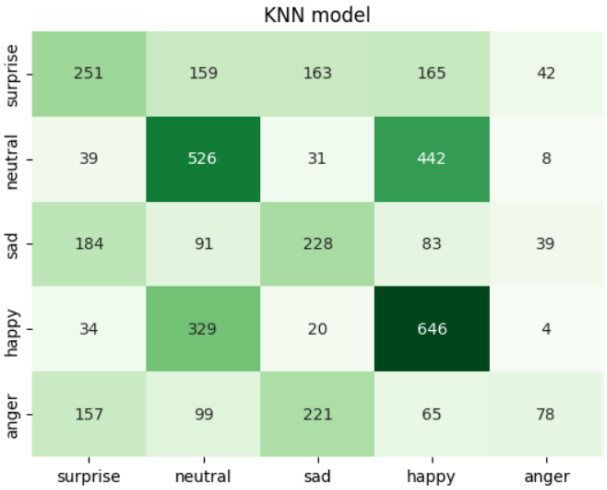


Рисунок 16 – Матрица предсказаний KNN

Мы видим, что KNN плохо справилась с определением эмоций, и лишь одна эмоция радости смогла быть определена выше порога 0,5. Эмоцию злости определить оказалось сложнее всего f1 = 0,29.

Модель Random forest на тестовой выборке получились результаты, изображенные на рисунке 17, а для обучающей на рисунке 18. Матрица предсказаний изображена на рисунке 19.

Random Forest model classfication report

	precision	recall	f1-score	support
surprise	0.53	0.60	0.56	780
neutral	0.74	0.86	0.80	1046
sad	0.51	0.31	0.39	625
happy	0.86	0.82	0.84	1033
anger	0.57	0.60	0.59	620
accuracy			0.68	4104
macro avg	0.64	0.64	0.63	4104
weighted avg	0.67	0.68	0.67	4104

Рисунок 17 – Результаты Random forest на Test

Random Forest model classfication report

	precision	recall	f1-score	support
surprise	1.00	1.00	1.00	3259
neutral	1.00	1.00	1.00	4080
sad	1.00	1.00	1.00	2466
happy	1.00	1.00	1.00	4011
anger	1.00	1.00	1.00	2598
accuracy			1.00	16414
macro avg	1.00	1.00	1.00	16414
weighted avg	1.00	1.00	1.00	16414

Рисунок 18 – Результаты Random forest на Train

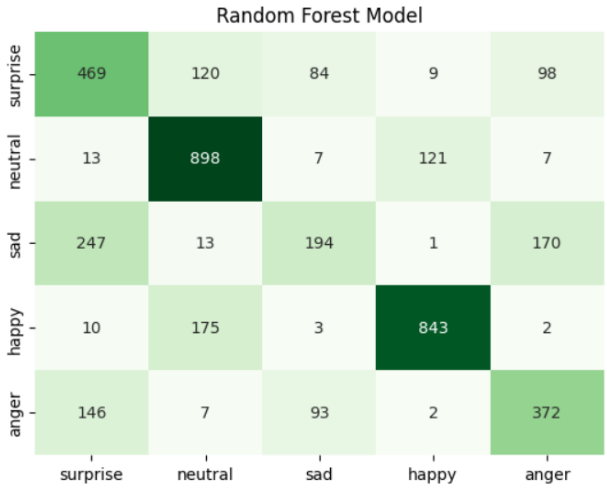


Рисунок 19 – Матрица предсказаний Random forest

Случайный лес показывает неплохие результаты, хорошо классифицирует эмоции радости и нейтральности. Лучше броска монетки угадывает злость и удивление, но не грусть. Явно видно переобучение модели на тренировочной выборке.

И наконец, для модели CNN на тестовой выборке получились результаты, изображенные на рисунке 20, а для обучающей на рисунке 21. Матрица предсказаний изображена на рисунке 22.

```
129/129 [==============================] - 30s 236ms/step
CNN model classification report on test set:
              precision    recall  f1-score   support

    surprise       0.67      0.70      0.69       780
     neutral       0.86      0.82      0.84      1046
         sad       0.58      0.53      0.56       625
       happy       0.90      0.89      0.90      1033
       anger       0.60      0.67      0.64       620

    accuracy                           0.75      4104
   macro avg       0.72      0.72      0.72      4104
weighted avg       0.75      0.75      0.75      4104
```

Рисунок 20 – Результаты CNN на Test

```
513/513 [==============================] - 123s 239ms/step
CNN model classification report on training set:
              precision    recall  f1-score   support

    surprise       0.99      1.00      0.99      3259
     neutral       1.00      0.99      0.99      4080
         sad       0.99      0.99      0.99      2466
       happy       1.00      1.00      1.00      4011
       anger       0.99      0.99      0.99      2598

    accuracy                           0.99     16414
   macro avg       0.99      0.99      0.99     16414
weighted avg       0.99      0.99      0.99     16414
```

Рисунок 21 – Результаты CNN на Train

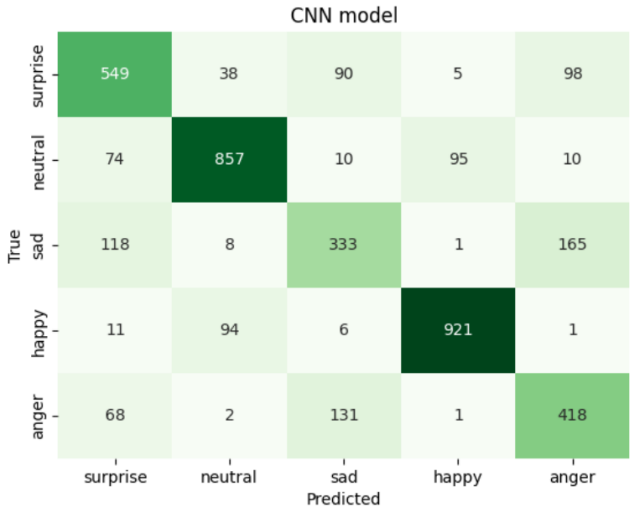


Рисунок 22 – Матрица предсказаний CNN

CNN состоящая из 7 слоев показала отличный результат accuracy = 0.75. Мы видим, как хороша она справилась с задачей и все эмоции прошли порог угадывания.

Так же была произведена проверка на независимых данных и результат показан на рисунке 23.

```
133/133 [==============================] - 37s 58ms/step - loss: 0.3601 - accuracy: 0.9189
Loss on personal dataset set: 0.3601000931
Accuracy on personal dataset set: 0.918944213
```

Рисунок 23 – результат CNN на независимых данных

Как мы видим точность предсказаний 0.91, что является отличным результатом.

И наконец, AlexNet. Результаты на тестовых данных изображены на рисунке 24, а на обучающей выборке на рисунке 25. Так же была произведена проверка на независимых данных и результат показан на рисунке 26.

```
Alex Net test Report:
              precision    recall  f1-score   support

    surprise       0.00      0.00      0.00       780
     neutral       0.25      1.00      0.41      1046
         sad       0.00      0.00      0.00       625
       happy       0.00      0.00      0.00      1033
       anger       0.00      0.00      0.00       620

    accuracy                           0.25      4104
   macro avg       0.05      0.20      0.08      4104
weighted avg       0.06      0.25      0.10      4104
```

Рисунок 24 – Результаты Alex Net на Test

```
Alex Net train Report:
              precision    recall  f1-score   support

    surprise       0.00      0.00      0.00      3259
     neutral       0.25      1.00      0.40      4080
         sad       0.00      0.00      0.00      2466
       happy       0.00      0.00      0.00      4011
       anger       0.00      0.00      0.00      2598

    accuracy                           0.25     16414
   macro avg       0.05      0.20      0.08     16414
weighted avg       0.06      0.25      0.10     16414
```

Рисунок 25 – Результаты Alex Net на Train

```
133/133 [==============================] - 3s 22ms/step - loss: 1.5156 - accuracy: 0.2253
Loss on personal dataset set: 1.5156214633211
Accuracy on personal dataset set: 0.2253131469212
```

Рисунок 26 – Результаты Alex Net на независимых данных

Такие результаты связаны с тем, что Alex Net обучался на данных размера 256 × 256, а у нас данные размера 96 x 96 притом еще и цветные. Как мы видим, сеть игнорировала все настроения кроме нейтральности.

В таблице 1 приведены значения метрик для нейронных сетей.

Таблица 1. Полученные метрики для нейронных сетей

	CNN	Alex Net
Accuracy	0.75	0.25
Precision	0.722	0.08
Recall	0.722	0.2
F1	0.726	0.05
Accuracy Test	0.919	0.2253

В таблице 2 приведены значения метрик для других алгоритмов

Таблица 2. Полученные метрики для алгоритмов

	XGBoost	KNN	Random forest
Accuracy	0.72	0.42	0.68
Precision	0.696	0.416	0.64
Recall	0.69	0.388	0.64
F1	0.693	0.38	0.63

V. Заключение

Был рассмотрен датасет AffectNet на части которого происходило обучение и тестирование трех алгоритмов XGBoost, KNN, Random forest направленных на распознавание эмоций на лице человека по фотографии. Были рассмотрены две нейронные сети одна классическая CNN, вторая Alex Net. Каждая модель была рассмотрена с точки зрения ее архитектурных особенностей. Лучше всего себя проявила CNN в решении этой задачи, немного хуже показала себя на пару процентов XGBoost, на третьем месте оказалась модель случайного леса, на четвертом KNN показала плохие результаты, а на самом худшем месте оказалась Alex Net. Для KNN это объясняется различиями в обучающих процессах, так как фотографии лиц содержат различные шумы и изменения в данных такие как изменения освещения или угла съемки,

так же эмоции на лицах порой сложно классифицировать. Данный фактор усложняет определение ближайших соседей.

Для Alex Net это объяснимо другими данными на которых происходило обучение.

Алгоритмы в которых использовались деревья решений показали себя хорошо, так как данные алгоритмы хорошо справляются с нелинейными зависимостями, а также устойчивы к шуму.

CNN показала лучшие результаты, так как такие алгоритмы имеют специальную архитектуру, которая способна изучать пространственные признаки и способны к изучению объектов, которые были сдвинуты или повернуты, что позволяет CNN быть одним из лучших инструментов для классификации изображений.

Литература

[1] D. B. Pazychev, K. S. Bakulev and R. N. Sadekov, "Low-CostNavigation System for UAV," 2023 30th Saint PetersburgInternationalConference on Integrated Navigation Systems (ICINS),SaintPetersburg, Russian Federation, 2023, pp. 1-6,doi:10.23919/ICINS51816.2023.10168469.

[2] Ali, B., Sadekov, R.N., Tsodokova, V.V., A Review of NavigationAlgorithms for Unmanned Aerial Vehicles Based on Computer VisionSystems, Gyroscopy and Navigation, 2023

[3] Guzhva, N.S., Prun, V.E., Postnikov, V.V., Sadekov, R.N., Sholomov,D.L., Using 3D Object Detection DNN in an Autonomous Tram toPredict the Behaviour of Vehicles in the Road Scene, 29th SaintPetersburg International Conference on Integrated Navigation Systems,ICINS 2022

[4] Guzhva, N.S., Ali, B., Bakulev, K.S., Sadekov, R.N., Sholokhov, A.V.Evaluating the Accuracy of Tram Positioning System in High-RiseBuilding Environment Using Data from Visual GeoinformationSystems, 30th Anniversary Saint Petersburg International Conferenceon Integrated Navigation Systems, ICINS 2023, 2023

[5] Компьютерное зрение видит эмоции, пульс, дыхание и ложь — но как построить на этом стартап. Разговор с Neurodata Lab // https://habr.com/ru/companies/habr_career/articles/465153/

[6] T. Hussain, A. Iqbal, B. Yang, A. Hussain, “Real time violence detection in surveillance videos using Convolutional Neural Networks” ResearchGate 2022

[7] Искусственный интеллект поможет машинам распознавать эмоции // 14.07.2023 // https://www.osp.ru/articles/2023/0714/13057346?ysclid=lv28penlen406681644

[8] Stephen Chen // ‘Forget the Facebook leak’: China is mining data directly from workers’ brains on an industrial scale // 29 Apr 2018 // https://www.scmp.com/news/china/society/article/2143899/forget-facebook-leak-china-mining-data-directly-workers-brains?campaign=2143899&module=perpetual_scroll_0&pgtype=article

[9] Ali Mollahosseini, Student Member, IEEE, Behzad Hasani, Student Member, IEEE, and Mohammad H. Mahoor, Senior Member, IEEE. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild

[10] https://www.kaggle.com/datasets/noamsegal/affectnet-training-data/data

Исследование возможности детектирования поддонов с грузами

М. Н. Шаталов
кафедра инженерной кибернетики
НИТУ «МИСиС»
Москва, Россия
m2107866@edu.misis.ru

Аннотация — в данной работе рассматривается проблема автоматического обнаружения поддонов в промышленных условиях с помощью одной камеры RGB. Эта проблема актуальна в контексте автономной навигации транспортных средств, а также для таких задач хранения и извлеченя поддонов. В частности, представлен подход, основанный на сверточной нейронной сети (CNN), за которой следует этап принятия решения. Сверточная нейронная сеть обучается распознавать два элемента поддона, а именно лицевую сторону поддона и его карманы. Также произведено сравнение трех современных CNN: Faster R-CNN, SSD и YOLOv4. Для обучения и оценки на складе был собран набор данных. Набор данных содержит изображения поддонов в различных конфигурациях, как на земле, так и на стеллажах, и с произвольной ориентацией. В целом, результаты показывают, что более быстрые R-CNN и SSD работают лучше, чем YOLOv4.

Ключевые слова — Компьютерное зрение, Детекция поддонов, Распознавание поддонов, Складские помещения, YOLO, R-CNN, SSD

I. Введение

За последнее десятилетие глубокое обучение значительно улучшило скорость и точность обнаружения объектов по сравнению с традиционными методами компьютерного зрения. В этой работе рассматривается проблема автоматического обнаружения поддонов в промышленных условиях, используя сверточные нейронные сети (CNN) и одну RGB-камеру. Поддон — это деревянная конструкция, используемая для перемещения или хранения товаров. В каждом поддоне есть два кармана, то есть два отверстия, в которые необходимо вставлять вилы вилочного погрузчика. Цель задачи - идентифицировать все поддоны в заданной рамке изображения, попытаться определить их размер и отсканировать QR-код. Идентификация поддонов важна в нескольких случаях промышленного использования автономных управляемых транспортных средств (AGVs), и которые требуют понимания сцены в неструктурированных средах [1]. Примерами применения являются предотвращение столкновений роботизированных вилочных погрузчиков, снятие поддона со стеллажа или хранение поддона в заданном месте.

В работе предлагается пайплайн, который сначала определяет границы соответствующих элементов поддона с помощью CNN, а именно лицевую сторону поддона и его карманы. После этого применяется блок принятия решений, который использует эти элементы для выбора окончательных предложений по поддону. В частности, для обеспечения возможности обнаружения поддона вблизи датчика необходимо, чтобы была видна как лицевая сторона поддона, так и два его кармана. Этот консервативный подход направлен на обеспечение безопасности и надежности работы AGV. Кроме того, в работе представлено сравнительное исследование, оценивающее производительность трех глубоких нейронных сетей: более быстрой R-CNN, SSD и YOLOv4.

Одним из основных недостатков сверточных нейронных сетей является необходимость в большом количестве аннотированных выборок [2]. Обычно в исследовательских работах CNN обучаются и оцениваются на основе стандартных наборов данных. Однако приложения промышленного компьютерного зрения имеют специфические потребности, такие как обнаружение объектов, недоступных в стандартных наборах данных (например, поддонов). Кроме того, приложения промышленного компьютерного зрения должны адаптироваться к различным условиям окружающей среды [3]. Поэтому в этой работе также представлен новый набор данных RGB-изображений, который был получен на промышленном объекте (складе). Изображения набора данных содержат поддоны различной конфигурации, т.е. изображения могут содержать несколько поддонов, расположенных на разной высоте, как на земле, так и на стеллажах. Кроме того, поддоны могут иметь произвольную ориентацию. Кроме того, поддоны могут быть частично покрыты прозрачной пластиковой пленкой.

Статья организована следующим образом. В разделе II рассматривается современное состояние автоматического обнаружения поддонов в промышленных условиях. В разделе III описывается предлагаемый метод, а в разделе IV - набор данных. В разделе V представлены результаты эксперимента. Раздел VI завершает работу.

II. Состояние индустрии

Работ по автоматическому обнаружению поддонов при помощи моделей глубокого обучения в настоящее время не так уж и много. Несколько авторов исследовали традиционные подходы компьютерного зрения для автоматического обнаружения поддонов на основе стереовидения и фронтального обзора объекта, [4], [5], [6]. Традиционные методы обработки изображений рассматривались в других подходах с использованием монокулярного зрения с дополнительным допущением наличия единственного поддона, расположенного на земле с двумя видимыми сторонами [7], [8], [9], [10]. Молтер и др. [11] исследовали использование time-of-flight(ToF) камер для детектирования поддонов, размещенных на земле.

В [12], [13] были представлены два метода, которые не являются устойчивыми к различным условиям освещения, поскольку они используют информацию о цвете для отделения палитры от фона. В [14] для автономных вилочных погрузчиков была разработана монокулярная система визуального наблюдения, которая распознает поддон по прямоугольным элементам. Проблема

Рис 1 – рассматриваемый пайплайн для автоматического детектирования поддонов

автоматического обнаружения паллета была также изучена с использованием и других типов датчиков или способов обработки данных. В [15], [16] обнаружение поддонов было достигнуто за счет использования плоских лазерных сканеров.

III. ПРЕДЛАГАЕМЫЙ МЕТОД

Алгоритм предлагаемого метода автоматического обнаружения поддонов показан на рисунке 1. Каждое входное изображение сначала обрабатывается CNN, которая обнаруживает две категории объектов, а именно: двумерную ограничивающую рамку лицевой стороны поддона и двумерную ограничивающую рамку кармана поддона. Центр кармана, ограничивающий рамку, представляет собой ценную информацию, которая может быть использована для определения целевой конфигурации двух вилок вилочного погрузчика для задачи извлечения поддонов. После этого выполняется блок принятия решения, который применяет эвристические правила для выбора окончательных действий в отношении паллет.

A. Архитектуры сверточных нейронных сетей

Для решения задачи обнаружения поддонов было проведено сравнение трех архитектур сверточных нейронных сетей: Faster R-CNN [19], Single Shot Multi-box Detector (SSD) [20] и You Only Look Once v4 (YOLOv4) [21]. Faster R-CNN использует двухэтапный подход, также называемый region-based: первая часть сети выполняет поиск объектов внутри изображения независимо от класса, а затем классифицирует их во второй части. В R-CNN был принят алгоритм выборочного поиска для обнаружения гипотез расположения регионов на входном изображении. Впоследствии, в 2015 году, Girshick представила Fast R-CNN, которая добилась лучшей производительности с точки зрения времени вычислений, а также большей точности за счет генерации гипотез расположения регионов непосредственно на карте объектов, вычисленной по всему изображению. Следовательно, больше не было необходимости для каждой идентифицированной ограничивающей рамки на изображении проходить процедуру извлечения объектов CNN. Однако, Fast R-CNN по-прежнему использовала алгоритм выборочного поиска, который был узким местом. В 2017 году Рен и др. представили Faster R-CNN [19], где алгоритм выборочного поиска был заменен Region Proposal Network (RPN). RPN - это CNN для выявления гипотез по регионам.

Таким образом, Faster R-CNN использует полностью сверточную сеть для извлечения признаков и формирования карты признаков входного изображения. Сеть для извлечения признаков может быть разных типов (VGG, ResNet, Inception и т.д.). В этой работе основой для более быстрого R-CNN является ResNet50. Карта признаков используется RPN для прогнозирования гипотез по регионам. После этапа ROI pulling, который помогает стандартизировать гипотезы, классификатор определяет класс объекта.

В этой работе сеть Faster R-CNN была обучена на 10000 итераций с размером батча, равным 2 . Скорость обучения была установлена равной 0,0025, при этом начальный прогон содержал 200 итераций. Размер входных изображений был изменен с сохранением соотношения сторон, так что самая короткая сторона каждого изображения имела случайное значение среди (640, 672, 704, 736, 768, 800) пикселей. Максимальный размер стороны составлял 1333 пикселя. Время обучения заняло около 12 часов.

В то время как Faster R-CNN основана на двухэтапном подходе, SSD и YOLO предлагают одноэтапную сеть обнаружения объектов. В частности, SSD не требует указания региона. Методика SSD заключается в том, что изображение проходит определенное количество сверток для извлечения признаков, в результате чего получается слой признаков размером m x n (количество местоположений) с p каналами, например, 8 x 8 или 4 x 4 (см.рис. 3). К этому слою признаков размером m x n x p применяется свертка размером 3 x 3. Для каждого местоположения получается k ограничительных рамок. Эти k ограничительных рамок имеют разные размеры и пропорции. Для каждой ограничительной рамки вычисляется с оценок класса и 4 смещения относительно исходной формы ограничительной рамки по умолчанию. Таким образом, получается (c + 4)kmn выходов. В этой работе обучение SSD длилось более 150 эпох. Каждая эпоха проходит по всему обучающему набору один раз. Изначально размер изображений в наборе данных был изменен до фиксированного разрешения 512 × 512 пикселей. Размер пакета был установлен равным 8, скорость обучения - 0,004, коэффициент затухания - 0,0001, а импульс - 0,9. Обучение заняло около 8 часов.

Третьей сверточной нейронной сетью, используемой в предлагаемом сравнении, является YOLOv4. YOLO была представлена в своей первой версии в 2016 году Редмоном и соавторами [21]. В YOLO каждое входное изображение разбивается на сетку, и для каждой ячейки определяется определенное количество ограничивающих рамок. Несмотря на быстрое обнаружение, ошибка локализации в YOLO не была незначительной. YOLOv2, также называемая YOLO 9000, улучшила точность обнаружения. Она была основана на Darknet-19. Darknet - это быстрый фреймворк для нейронных сетей, написанный на C и CUDA. Однако одним из главных недостатков было обнаружение мелких предметов.

Затем в качестве дополнительного улучшения была представлена YOLOv3. Он был основан на Darknet-53. Затем Бочковский и др. представили YOLOv4 [21], которая повышает точность и скорость работы по сравнению с YOLOv3. Эта версия YOLO нацелена на создание точной и быстрой сети, которую можно использовать для обучения на одном обычном графическом процессоре. Авторы провели несколько экспериментов по оптимизации компонентов архитектуры CNN. Кроме того, они внесли дополнительные улучшения в виде добавления модулей мозаики, Self-Adversarial Training (SAT) и Cross mini-Batch Normalization (CmBN). Основой является CSPDarknet53. В этой работе исходные изображения для YOLOv4 были изначально изменены на 608×608. Обучение проходило в течение 24000 итераций с размером батча в 16 циклов. Это заняло около 23 часов. Скорость обучения была установлена равной 0,001, с затуханием 0,0005 и импульсом 0,949.

B. Блок принятия решений

После оценки CNN обнаруженные лицевые стороны поддонов и карманы для поддонов используются для выбора определения дальнейших действий путем применения блока принятия решения. На этапе принятия решения используются следующие эвристические правила. Если обнаруженная лицевая сторона поддона превышает пороговое значение по площади, предложение по поддону создается только в том случае, если на лицевой стороне поддона имеется ровно два кармана, в противном случае оно отбрасывается. В этом случае, в предложение поддона включаются лицевая сторона поддона и два кармана на ней. Порог области был установлен на уровне 150 × 103 пикселей. Это правило принятия решений продиктовано тем фактом, что для безопасного выполнения операций по развальцовке паллет, расположенных вплотную друг к другу и приблизительно расположенных фронтально, важно распознавать все компоненты паллет. Более целесообразно не воспринимать поддон, даже если он действительно присутствует (ложноотрицательный результат), чем генерировать ложноположительный результат, который может привести к опасным ситуациям, таким как столкновение погрузчика с окружающей средой. Таким образом, можно было бы использовать логистическую функцию для двух классов, которая имеет следующий вид:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \qquad (1)$$

Однако, для возможного расширения количества классов детектируемых объектов было принято решение воспользоваться функцией SoftMax, которая имеет следующий вид:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \qquad (2)$$

Если обнаруженная лицевая сторона поддона меньше порогового значения площади, блок принятия решений всегда принимает его, как поддон, независимо от того, есть ли в нем два кармана. Любой карман, расположенный на лицевой стороне поддона, включается в гипотезу по определению поддона. Это второе правило принятия решений обусловлено тем фактом, что для навигации погрузчика важно смоделировать все возможные поддоны. Ячейка считается расположенной внутри лицевой стороны поддона, если пересечение их ограничивающих рамок составляет не менее 80% площади ограничивающей рамки ячейки. Результаты экспериментов, проводимых в разделе 5, анализируются путем расчёта метрики Average Precision:

$$AP@K = \frac{1}{K}\sum_{k=1}^{K} r^{true}\left(\pi^{-1}(k)\right) * p@k \qquad (3)$$

Рис 2 – Примеры изображений из датасета

IV. НАБОР ДАННЫХ

Набор данных был собран на складе DIY ритейлера. Товарами на поддонах были различные товары от поставщиков. Была использована монокулярная RGB-камера (разрешение 3280×2464), подключенная к встроенной системе Jetson Nano. Камера была установлена на штативе с регулируемой высотой. Полученный набор данных содержит 1344 изображения и был разделен на обучающий набор (991 изображение) и тестовый набор (353 изображения). Изображения были сделаны на различных расстояниях от поддонов. Более того, поддоны были обнаружены в различных конфигурациях: сложенные штабелями, на полу, на стеллажах и с точки зрения вилочного погрузчика, выполняющего операцию разветвления.

Набор данных был сгенерирован при различном освещении, т.е. некоторые снимки были сделаны утром при естественном освещении, в то время как другие снимки были сделаны вечером, когда на складе преобладало искусственное освещение. Кроме того, в некоторых случаях использовался фонарик, чтобы осветить поддоны, расположенные рядом с камерой. На рисунке 2 показаны примеры изображений набора данных.

Несколько снимков были получены с одной и той же позиции камеры при разном освещении. Подгруппой называется набор снимков, сделанных с одной и той же позиции камеры. Некоторые снимки в рамках одной подгруппы были сделаны путем ослепления камеры

Рис 3 – Примеры изображений из одной подгруппы

Таблица I
Группы и подгруппы датасета.

Group name	Distance	Height	Number of subgroups
Group 1	150 cm	27.5 cm	68
Group 2	300 cm	27.5 cm	35
Group 3	250 cm	93 cm	24
Group 4	300 cm	151 cm	12
Group 5	300 cm	27.5 cm	12
Group 6	variable	95 cm	1

фонариком, чтобы сделать фон темнее, другие снимки были сделаны путем наведения камеры ближе к полу. Наконец, некоторые изображения были получены путем просвечивания прозрачной пластиковой пленки, которой были

обернуты поддоны. На рисунке 3 показаны примеры изображений, полученных из одной и той же подгруппы. Информация о подгруппах изображений представлена в таблице I. Подгруппы были объединены в группы. Группа содержит все подгруппы, которые имеют одинаковое расстояние от камеры до ближайшего поддона на изображении, а также одинаковую высоту камеры от земли.

V. ЭКСПЕРИМЕНТЫ

A. Результаты детектирования сетями

Полученный набор данных был использован для обучения трех сетевых архитектур. Набор данных был расширен с использованием традиционных методов для повышения обобщения и надежности для многих возможных ситуаций на складе. Применяемые методы увеличения объема данных включают горизонтальное перемещение и случайную обрезку, случайное вращение, случайное изменение яркости. Был применен метод трансферного обучения. В частности, обучение началось с модели, предварительно подготовленной на базе набора данных MS COCO [22]. Кроме того, были доработаны некоторые гиперпараметры, чтобы максимально повысить производительность. Для обучения использовался графический процессор Nvidia GeForce GTX 1080 Ti. Результаты были собраны и проанализированы с помощью COCO metrics.

Как показано в таблице III, промежуточные результаты использованного пайплайна показывают, что CNN,

Таблица II
COCO метрики для трех выбранных нейросетей

	Faster R-CNN	SSD	YOLOv4
AP	75.4	75.8	69.1
AP_{50}	93.8	92.0	86.3
AP_{75}	89.7	88.0	82.1
AP_S	0.0	6.5	14.3
AP_M	37.7	40.5	18.3
AP_L	78.8	78.3	73.6
AR_{max1}	12.6	12.6	11.7
AR_{max10}	53.3	53.6	49.4
AR_{max100}	82.1	80.4	78.0
AR_S	0.0	6.3	18.4
AR_M	53.4	50.9	31.1
AR_L	84.8	82.7	82.1
Pallet front side AP	80.1	81.5	69.5
Pocket AP	70.8	70.1	68.7

которые достигли наивысших значений средней точности (AP) — это Faster R-CNN и SSD. Тем не менее, YOLOv4 добилась лучшего обнаружения объектов, наблюдаемых на большом расстоянии. В COCO metrics объекты считаются маленькими, если их площадь ограничивающего прямоугольника меньше, чем 32 × 32 пикселя. Тестовый набор содержит 46 небольших объектов (только карманы поддонов), наблюдаемых на большом расстоянии, то есть менее 1% от общего числа тестовых экземпляров. Faster R-CNN не смогла обнаруженить мелкие объекты, поскольку AR_S (Average recall) и AP_S (Average precision) равны 0. В целом, можно заметить, что AP выше для категории "лицевая сторона поддона", чем для категории "карман для поддона ". Этот результат можно объяснить тем, что лицевая сторона поддона имеет более характерные особенности, чем карманы для поддонов. Карманы для поддонов обычно выглядят как темные области, но иногда они становятся менее темными, когда камера видит задний план.

С точки зрения времени вычисления Faster-RCNN может обрабатывать изображения со скоростью 15 кадров в секунду, в то время как SSD работает со скоростью 35,44 кадра в секунду, а YOLOv4 – со скоростью 27,8 кадров в секунду. Однако в промышленных приложениях для беспилотных ТС потребность в повышении эффективности обработки изображений не так важна, как необходимость в точности операций, например, по взаимодействию с поддонами и безопасности навигации. Некоторые результаты, полученные с помощью Faster R-CNN, показаны на рисунке 4. Обнаруженные объекты ограничены рамками, ориентированными по оси. Можно видеть, что лицевые стороны поддонов, расположенных рядом с камерой, распознаются правильно. Более того, система обнаружения карманов надежна даже при наличии обернутых поддонов.

B. Блок принятия решений

В таблице IV представлены результаты обнаружения поддонов после выполнения блока принятия решения в конце предлагаемого пайплайна. В предлагаемой задаче оценка AP_{75} (средняя точность для объектов с пересечением, превышающим порог объединения 0,75) особенно важна, поскольку это строгий показатель, который полезен, когда требуется высокая точность. В целом,

Рис 4 – Примеры определения лицевых сторон и карманов поддонов с использованием Faster R-CNN

эксперименты подтверждают, что Faster R-CNN и SSD работают лучше, чем YOLOv4. Более того, Faster R-CNN дает несколько лучшие результаты, чем SSD. Также можно заметить, что средняя точность для крупных объектов COCO (показатель AP_L) значительно выше, чем для средних объектов COCO (показатель AP_M), для всех трех CNN. Средняя точность и полнота определения мелких объектов COCO (показатели AP_S и AR_S) отрицательны, поскольку все мелкие объекты были удалены блоком принятия решений.

VI. ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен метод автоматического обнаружения поддонов, основанный на монокулярной системе обзора. Было проведено сравнение трех сверточных нейронных сетей на основе оригинального набора данных, собранного на складе. Результаты экспериментов показывают, что Faser R-CNN и SSD работают лучше, чем YOLOv4.

Таблица III

Метрики определения паллет после применения блока решений

	Faster R-CNN	SSD	YOLOv4
AP	78.4	77.9	72.0
AP_{50}	93.2	90.5	87.7
AP_{75}	92.0	89.8	86.1
AP_S	-100.0	-100.0	-100.0
AP_M	28.7	17.9	14.0
AP_L	79.1	78.9	72.7
AR-$max1$	25.0	23.7	23.1
AR-$max10$	74.1	70.5	68.4
AR-$max100$	85.3	81.6	79.7
AR_S	-100.0	-100.0	-100.0
AR_M	33.4	20.2	19.6
AR_L	86.1	82.5	80.7

ЛИТЕРАТУРА

[1] N. Panokin, N.I. Kotov et al, “Computer vision system as an additional aid in car navigation”, ResearchGate, 2017.

[2] R. Sadekov, Dmitry Pazychev et al, “Low-Cost Navigation System for UAV”, ResearchGate, 2023

[3] A. Bushra, R. Sadekov et al, “A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems”, ReseachGate, 2023

[4] T. Li, B. Huang, C. Li, and M. Huang, “Application of convolution neural network object detection algorithm in logistics warehouse,” The Journal of Engineering, vol. 2019, no. 23, pp. 9053–9058, 2019.

[5] L. Sabattini, M. Aikio, P. Beinschob, M. Boehning, E. Cardarelli, V. Digani, A. Krengel, M. Magnani, S. Mandici, F. Oleari, C. Reinke, D. Ronzoni, C. Stimming, R. Varga, A. Vatavu, S. Castells Lopez, C. Fantuzzi, A. Mayr ¨ a, S. Nedevschi, C. Secchi, and K. Fuerstenberg, ¨ “The PAN-Robots Project: Advanced Automated Guided Vehicle Systems for Industrial Logistics,” IEEE Robotics Automation Magazine, vol. 25, no. 1, pp. 55–64, 2018.

[6] R. Varga and S. Nedevschi, “Robust Pallet Detection for Automated Logistics Operations,” in 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPH), 2016, pp. 470–477.

[7] R. Varga, A. Costea, and S. Nedevschi, “Improved autonomous load handling with stereo cameras,” in International Conference on Intelligent Computer Communication and Processing (ICCP), 2015, pp. 251–256.

[8] R. Varga and S. Nedevschi, “Vision-based autonomous load handling for automated guided vehicles,” in 10th International Conference on Intelligent Computer Communication and Processing (ICCP), 2014, pp. 239–244.

[9] M. Seelinger and J.-D. Yoder, “Automatic Pallet Engagment by a Vision Guided Forklift,” in IEEE International Conference on Robotics and Automation (ICRA), 2005, pp. 4068–4073.

[10] J. Pages, X. Armangu ` e, J. Salvi, J. Freixenet, and J. Marti, “A Computer ´ Vision System for Autonomous Forklift Vehicles in Industrial Environments,” in 9th Mediterranean Conference on Control and Automation (MEDS), 2001.

[11] Young Hun Song, Jee Hun Park, Suk Lee, and Kyung Chang Lee, “Implementation of distributed architecture based on CAN networks for unmanned forklift,” in 37th Annual Conference of the IEEE Industrial Electronics Society (IECON), 2011, pp. 2595–2599.

[12] G. Chen, R. Peng, Z. Wang, and W. Zhao, “Pallet Recognition and Localization Method for Vision Guided Forklift,” in 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2012, pp. 1–4.

[13] J.-L. Syu, H.-T. Li, J.-S. Chiang, C.-H. Hsia, P.-H. Wu, C.-F. Hsieh, and S.-A. Li, “A computer vision assisted system for autonomous forklift vehicles in real factory environment,” Multimedia Tools and Applications, vol. 76, 11 2016.

[14] M. R. Walter, S. Karaman, E. Frazzoli, and S. Teller, “Closed-loop pallet manipulation in unstructured environments,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 5119–5126.

[15] Y. Kita, R. Takase, T. Komuro, N. Kato, and N. Kita, “Detection and localization of pallets on shelves using a wide-angle camera,” in 19th International Conference on Advanced Robotics (ICAR), 2019, pp. 785–792.

[16] L. Baglivo, N. Biasi, F. Biral, N. Bellomo, E. Bertolazzi, M. D. Lio, and M. D. Cecco, “Autonomous pallet localization and picking for industrial forklifts: a robust range and look method,” Measurement Science and Technology, vol. 22, no. 8, 2011.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE

Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in European Conference on Computer Vision (ECCV), 2016.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” ArXiv, vol. abs/2004.10934, 2020.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation (CVPR),” in IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[21] R. Girshick, “Fast R-CNN,” in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.

[22] P. Chao, C. Kao, Y. Ruan, C. Huang, and Y. Lin, “Hardnet: A low memory traffic network,” in IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 3551–3560.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in con- ´ text,” in Computer Vision – ECCV 2014, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.