

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК
Руководитель образовательной программы
09.03.04 Программная инженерия

(подпись) И. Н. Обабков
(Ф.И.О.)
«24» _____ мая 2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
Пояснительная записка

**РАЗРАБОТКА МЕТОДА ВЫДЕЛЕНИЯ ОБЪЕКТОВ НА
ТЕПЛОВИЗИОННЫХ АЭРО-ФОТОСНИМКАХ**

Руководитель, канд. физ.-мат. наук	_____ И. С. Стародубцев
Нормоконтроль	_____ А. Т. Маканова
Обучающийся группы РИ-400013	_____ С. В. Вяцков

Екатеринбург 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»
Институт радиоэлектроники и информационных технологий – РТФ

Школа бакалавриата

Направление (специальность) 09.03.04 Программная инженерия

УТВЕРЖДАЮ

Руководитель образовательной программы
09.03.04 Программная инженерия

И. Н. Обабков

(подпись)

(Ф.И.О.)

«11» _____ декабря 2023 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

обучающегося Вяцкова Семена Валерьевича группы РИ-400013

(фамилия, имя, отчество)

1 Тема выпускной квалификационной работы

Разработка метода выделения объектов на тепловизионных аэро-фотоснимках

Утверждена распоряжением по институту от «04» декабря 2023 г. № 33.02-05/298

2 Руководитель Стародубцев Илья Сергеевич, канд. физ.-мат. наук, доцент кафедры ИИТ

(Ф.И.О., ученая степень, ученое звание, должность)

3 Исходные данные к работе

Видеоматериалы, представляющие записи тестовых полетов, нормативная, учебная, методическая литература по теме ВКР, материалы, полученные в ходе преддипломной практики, техническая документация, требования к системе от заказчика.

4 Содержание пояснительной записки (перечень подлежащих разработке вопросов)

1. Исследование предметной области

1.1 Методы компьютерного зрения

1.2 Инструмент аннотации изображений CVAT

1.3 Задача сегментации изображения

1.4 Задача классификации и обнаружения изображения

1.5 Окончательный выбор нейросети

2. Проектирование метода

3. Работа с предоставленными данными

5 Перечень демонстрационных материалов

Техническое задание на проектирование, презентация, дополнительные материалы к ВКР.

6 Календарный план

Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
Обзор литературы	до 15.01.2024 г.	
Анализ предметной деятельности	до 05.02.2024 г.	
Разработка и тестирование	до 25.04.2024 г.	
Выводы и заключение	до 10.05.2024 г.	

Руководитель _____
(подпись)

Стародубцев Илья Сергеевич
(Ф.И.О.)

Задание принял к исполнению 12.12.2023 г.
(дата)

(подпись обучающегося)

7 Выпускная квалификационная работа закончена «2» мая 2024 г.

Пояснительная записка и все материалы просмотрены

Считаю возможным допустить Вяцкова Семена Валерьевича к защите его выпускной квалификационной работы в Государственной экзаменационной комиссии.

Руководитель _____
(подпись)

8 Допустить Вяцкова Семена Валерьевича к защите выпускной квалификационной работы в Государственной экзаменационной комиссии (распоряжение по институту № 33.02-05/133 от «24» мая 2024 г.)

Руководитель образовательной программы
09.03.04 Программная инженерия _____
(подпись)

И. Н. Обабков
(Ф.И.О.)

РЕФЕРАТ

Выпускная квалификационная работа бакалавра 52 с., 14 рис., 64 источника.

РАЗРАБОТКА МЕТОДА, НЕЙРОСЕТИ, СОЗДАНИЕ ВЕСОВ НЕЙРОСЕТИ, АНАЛИЗ ТРЕБОВАНИЙ, ПРОЕКТИРОВАНИЕ МЕТОДА

Объект исследования – процесс обнаружения объектов.

Цель работы – анализ требований и проектирование метода для обнаружения объектов на аэро-фотоснимках.

Методы исследования: генерации идей, активизации творческого мышления, сравнений, абстрактно-логический, анализ и синтез, наблюдение, интервьюирование.

Результат работы: спроектирован метод обнаружения объектов на аэро-фотоснимках.

Выпускная квалификационная работа выполнена в текстовом редакторе и представлена в электронном виде.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	6
ВВЕДЕНИЕ	9
1 Исследование предметной области	11
1.1 Методы компьютерного зрения	11
1.2 Инструмент аннотации изображений CVAT	13
1.3 Задача сегментации изображения	15
1.4 Задача классификации и обнаружения изображения	19
1.5 Окончательный выбор нейросети	28
2 Проектирование метода	35
3 Работа с предоставленными данными	38
ЗАКЛЮЧЕНИЕ	45
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	47

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

AAT	Anchor-Aided Training (Обучение с привязкой)
AP	Average Precision (Средняя точность)
BiC	Bidirectional Concatenation (Билинейная конкатенация)
CNN	Convolutional Neural Network (Сверточная нейронная сеть)
CRF	Conditional Random Fields (Условно-случайные поля)
CVAT	Computer Vision Annotation Tool (Инструмент разметки [для задачи] компьютерного зрения)
CVPR	Computer Vision and Pattern Recognition (Компьютерное зрение и распознавание шаблонов)
DC-NAS	Divide-and-Conquer Neural Architecture Search (Разделяй-и-властвуй нейронная архитектура поиска)
DPM	Deformable Parts Model (Модель деформируемых деталей)
DevOps	Development & Operations (Разработка и операции)
EfficientDet	Efficient Detection (Эффективное обнаружение)
FAIR	Facility for Antiproton and Ion Research (Центр по исследованию ионов и Антипротонов)
FC-DenseNet	Fully Convolutional Densely Network (Полностью сверточная ёмкая сеть)
FCN	Fully Convolutional Network (Полностью сверточная сеть)
FPN	Feature Pyramid Networks (Сеть пирамиды особенностей)
FPS	Frames Per Second (Кадров в секунду)
G-RCNN	Granulated Region [Based] Convolutional Neural Network (Гранулированная региональная сверточная

	нейронная нейросеть)
GAN	Generative Adversarial Networks (Генеративно-сопоставительные сети)
HIT-UAV	High-altitude Infrared Thermal [dataset for] Unmanned Aerial Vehicle-based [object detection] (Высотный термовизионный [датасет для] беспилотных воздушных судов [для обнаружения объектов])
HOG	Histogram of Oriented Gradients (Гистограмма для ориентированных градиентов)
HRNet	High-Resolution Net (Сеть с высоким разрешением)
MS-COCO	Microsoft Common Objects in Context (Майкрософт рутинные объекты в контексте)
NAS-FPN	Neural Architecture Search [and] Feature Pyramid Networks (Нейронная архитектура поиска [и] нейросеть пирамиды особенностей)
PR-кривая	Precision-Recall-кривая (Кривая точности-полноты)
VOC	Visual Object Classes (Классы визуальных объектов)
RCNN	Region [Based] Convolutional Neural Network (Региональная сверточная нейронная сеть)
RNN	Recurrent Neural Network (Рекуррентная нейронная сеть)
SAM	Segment Anything Model (Сегментируй всё модель)
SPPNet	Single-Point Prompt Network (Нейросеть точечного шаблона)
SSD	Single-Shot Detector (Одноэтапная детекция)
SVM	Support Vector Machine (Машина опорных векторов)
UML	Unified Modeling Language (Унифицированный язык моделирования)
WSL	Windows Linux
YOLO	You Only Look Once (Вы смотрите только один раз)

k-NN	k-Nearest Neighbors (k ближайших соседей)
mAP	Mean Average Precision (Медиана средней точности)
УПКБ	Уральское Проектно-Конструкторское Бюро
MPT	Магнитно-Резонансная Томография
КТ	Компьютерная Томография

ВВЕДЕНИЕ

Сегодня всё больше цифровых методов существует для эффективной работы с данными, а данных, подходящих для обработки, существует великое множество. Отдельного упоминания заслуживают визуальные данные, которые могут обрабатываться методами компьютерного зрения.

Термовизионные аэро-снимки – это те данные, которые хорошо подходят для использования на них методов компьютерного зрения для обнаружения необходимых объектов, такова одна из задач программиста машинного обучения.

Налаживание обнаружения объектов для, пускай, термовизионных аэро-фотоснимков – это долгий и трудоемкий процесс, состоящий из технического задания, проектирования конвейера обнаружения объектов, согласование с заказчиком – на каждом этапе не исключены многочисленные правки. Чтобы ускорить этот процесс, используются уже созданные технологии программирования, которые можно подогнать под готовое решение, но даже в этом случае нужна работа программиста для адаптации под конкретную задачу. Процесс разработки может занимать несколько дней и недель.

Готовый метод обнаружения объектов на аэро-фотоснимках позволит составить унифицированный алгоритм, который в дальнейшем смогут использовать различные разработчики для конкретных своих задач.

Проблема состоит в потребности обнаружения объектов на аэро-фотоснимках.

Актуальность работы заключается в необходимости автоматизации процесса обнаружения объектов на термовизионных аэро-фотоснимках с целью эффективного формирования весов для нейросети.

Объект исследования – предоставленные мне данные, термовизионные аэро-фотоснимки.

Предмет исследования – процесс создания весов для нейросети.

Целью выпускной квалификационной работы является анализ требований и разработка метода обнаружения объектов на термовизионных аэро-фотоснимках.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- 1) провести анализ первоначальных требований к системе и предметной области;
- 2) провести сравнительный анализ существующий нейросетей, выявив функциональные возможности, а также недостатки существующих систем;
- 3) собрать требования для выявления потребностей системы обнаружения и формулирования на их основе функциональных и нефункциональных требований;
- 4) проанализировать требования к методу, вывив границы проекта, приоритеты задач, функциональные и нефункциональные требования;
- 5) разработать метод обнаружения объектов на термовизионных аэро-фотоснимках.

При выполнении работы использовались следующие методы: генерации идей, активизации творческого мышления, сравнений, абстрактно-логический анализ и синтез, наблюдение, интервьюирование.

Для выполнения выпускной квалификационной работы были использованы интернет-ресурсы на тему программирования, труды российских и зарубежных ученых по теме выпускной квалификационной работы, методическая литература, материалы, полученные в ходе преддипломной практики, техническая документация, требования к системе от заказчика.

Результатом работы является отработанный на тестовых данных метод обнаружения объектов на аэро-фотоснимках, метрики отработки тестовых данных, код дипломного проекта.

1 Исследование предметной области

1.1 Методы компьютерного зрения

Суть глубокого обучения заключается в запуске нейросети, которая выполняет задачу компьютерного зрения от начала до конца. Чаще всего непонятно, как сеть выдает определенный результат, и, следовательно, никогда нельзя понять, почему она дает сбой, когда это происходит. Объяснимость является проблемной областью исследований.

Классические конвейеры компьютерного зрения каждый раз разрабатываются вручную для конкретной задачи и представляют собой цикл предобработки, обработки, постобработки изображения с последующим получением полезной информации из картинки. Например, пример предобработки изображения – аффинное преобразование с сохранением параллельности прямых плоскости рисунка (рисунок 1). [1]

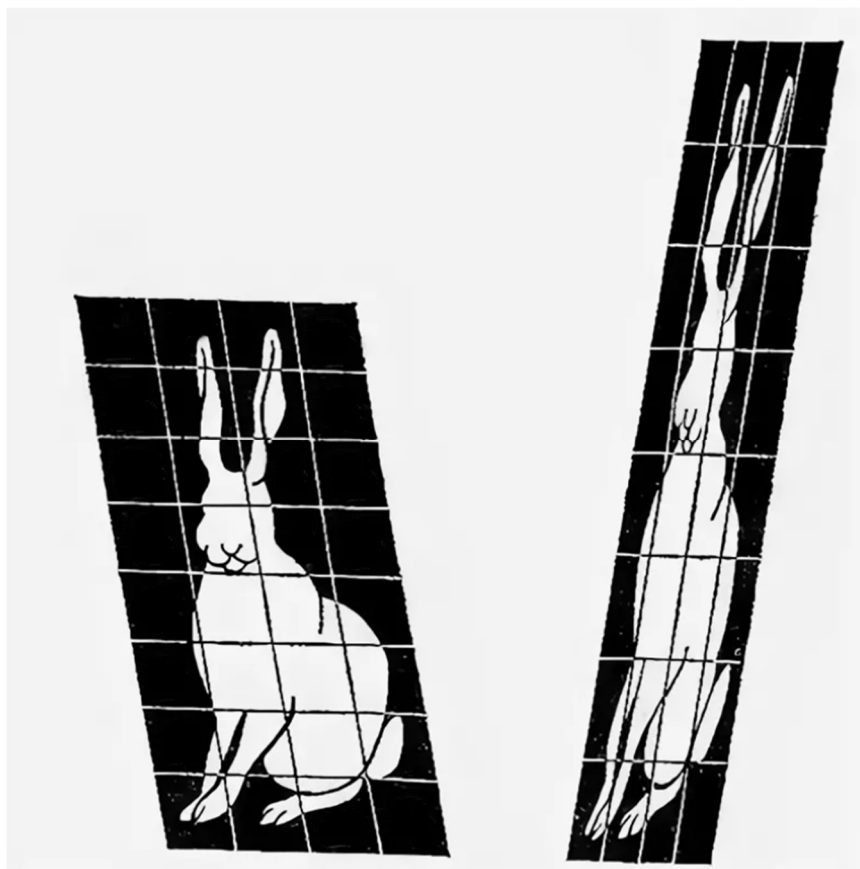


Рисунок 1 – Аффинное преобразование

Лучшие модели компьютерного зрения являются теми, которые сочетают в себе 2 парадигмы: использование классических конвейеров компьютерного зрения для упрощения или разбивки изображений на части, чтобы подготовить их к отправке в нейросеть глубокого обучения.

Классические конвейеры компьютерного зрения также иногда используются для контроля обучения нейросетей глубокого обучения, что позволяет создавать модели с неполным контролем там, где было бы дорого или невозможно получить достаточно большой набор обучающих данных.

Аналогичным образом, классические конвейеры компьютерного зрения иногда используются для получения синтетических данных для обучения нейросетей.

В качестве примера предположим, что создается конвейер компьютерного зрения, который фотографирует бумажные документы со смартфона и оцифровывает их. Можно было бы разработать модель нейросетей, которая выполняла бы задачу от начала до конца, но эта модель была бы сложной, дорогостоящей и ее было бы проблематично создавать и обучать.

Или можно объединить две парадигмы и создать меньший по размеру и более простой конвейер. Такая модель может быть разработана следующим образом:

- использовать нейросеть для определения местоположения документа (простой в получении набор данных, существующие модели обнаружения);
- использовать классическое компьютерное зрение для определения углов и трансформаций документа;
- использовать классическое компьютерное зрение для выполнения аффинных преобразований и исправления документа;
- использовать нейросеть для распознавания текста.

Дело в том, что редко возникает вопрос об использовании одного из них вместо другого. Ответ заключается в том, что почти всегда используйте и то,

и другое вместе для большинства реальных нишевых задач, в которых и возникает большая потребность.

Предпочтительно начинать с классического – правильно подобрать оптику, освещение и все физические компоненты. Максимально упростить задачу. Затем добавить обработку изображений, чтобы еще больше упростить задачу. Если в этот момент возможно выполнить это с желаемой точностью, лучше использовать классическое.

Лучше не прибегать к современным методам до тех пор, пока не будет найден тупик при решении задач классическим способом или пока они в основном не будут решены, но лучше всего добиваться повышения точности. До сих пор трудно понять, что пошло не так, когда нейросети неправильно классифицирует данные, и я хочу иметь возможность немедленно исправить неправильные классификации. Кроме того, редко есть уже имеющиеся данные, когда что-то нужно запустить в прямом эфире для подготовки надежного обнаружения с помощью нейросети – скорее всего, это будет верно, если вы решаете новую проблему, начиная с настройки камер. Если делается что-то, где можно загрузить миллионы образцов изображений с самого начала, возможно, лучше использовать нейросети.

1.2 Инструмент аннотации изображений CVAT

Для обучения моделей глубокого обучения, например, для обнаружения и распознавания объектов, требуются обширные коллекции изображений с метками соответствия действительности. Аннотация к изображению является процессом создания этих меток на изображениях из набора данных, который может быть использован для обучения модели (контролируемое обучение). Эти надписи содержат информацию о классах объектов, представленных на каждом изображении, их форме, местоположении и дополнительных атрибутах, таких как поза.

Инструменты аннотации изображений, такие как CVAT [2], облегчают создание изображений или видеокадров, создавая рабочие процессы, управляя классами и предоставляя формы (прямоугольники, многоугольники и т.д.) для указания точного местоположения классов. Такие инструменты для создания аннотаций могут быть запущены на локальном компьютере или в виде веб-инструментов для создания аннотаций, которые позволяют осуществлять совместную работу между членами команды.

CVAT расшифровывается как Computer Vision Annotation Tool; это бесплатный инструмент для цифровой анимации изображений с открытым исходным кодом, написанный на Python и JavaScript. CVAT поддерживает управляемые задачи машинного обучения для обнаружения объектов, классификации изображений, сегментации изображений и аннотирования 3D-данных.

Этот программный инструмент в последнее время приобрел большую популярность среди обычных и коммерческих пользователей. Следовательно, он также используется профессиональными группами по аннотированию данных для разработки наборов данных машинного обучения с контролем. Вы можете запускать CVAT практически на любой современной операционной системе (Ubuntu, Windows, Mac).

CVAT разрабатывается для аннотирования изображений с помощью компьютерного зрения, он разработан на основе отзывов от профессиональных групп по аннотированию данных, чтобы упростить аннотирование изображений для контролируемых задач машинного обучения.

Для обучения глубоких нейронных сетей, которые являются ядром машинного зрения, специалистам по обработке данных и компьютерному зрению требуется большое количество аннотированных данных. Изначально CVAT был разработан для внутреннего использования, чтобы обеспечить лучший метод крупномасштабного аннотирования тысяч изображений.

Этот процесс аннотирования очень трудоемок и занимает сотни или тысячи часов. Поэтому инструмент CVAT был разработан для ускорения

процесса аннотирования видео и изображений для использования в обучении алгоритмам компьютерного зрения.

1.3 Задача сегментации изображения

Компьютерное зрение на сегодняшний день достигло очень высокой точности, не меньше 90%. Ожидается, что технология будет усовершенствована до беспрецедентного уровня благодаря современным алгоритмам и методам сегментации изображений. Недавно лаборатория FAIR выпустила модель SAM [3], которая меняет правила сегментации изображений. Эта усовершенствованная модель может создавать подробные маски объектов из подсказок ввода, поднимая компьютерное зрение на новую высоту. Потенциально это может произвести революцию в том, как мы взаимодействуем с цифровыми технологиями в эту эпоху.

Сегментация изображения (рисунок 2) — это процесс в компьютерном зрении, который делит изображение на несколько областей или сегментов, каждый из которых представляет отдельный объект или область изображения. Этот подход позволяет экспертам изолировать определенные части изображения для получения значимой информации.

На рисунке 2 плоскость разбита на сегменты: движущиеся (MOVING) машины (CAR) и стационарные (STATIONARY) машины, тротуры (PAVEMENT), здания (BUILDING), дороги (ROAD), зелень (VEGETATION), забор (FENCE). [4]



Рисунок 2 – Сегментация изображения улицы

Модели сегментации изображений обучены улучшать качество вывода за счет распознавания важных деталей изображения и снижения сложности. Эти алгоритмы эффективно различают различные области изображения на основе таких характеристик, как цвет, текстура, контрастность, тени и края.

Сегментируя изображение, можно сосредоточить анализ на интересующих областях для получения более подробной информации. Ниже приведены различные методы сегментации изображений:

- 1) семантическая сегментация включает в себя маркировку пикселей по семантическим классам;
- 2) сегментация экземпляров идет дальше, обнаруживая и очерчивая каждый объект на изображении;
- 3) паноптическая сегментация присваивает уникальные идентификаторы экземпляров отдельным пикселям объекта, что приводит к более полной и контекстной маркировке всех объектов на изображении.

Сегментация реализуется с использованием моделей глубокого обучения на основе изображений. Эти модели извлекают все ценные данные и функции из обучающего набора. Затем можно превратить эти данные в векторы и матрицы, чтобы понять сложные функции. Вот некоторые из широко используемых моделей глубокого обучения, лежащих в основе сегментации изображений:

- 1) сверточные нейронные сети (CNN); [5]
- 2) полностью сверточные сети (FCN); [6]
- 3) рекуррентные нейронные сети (RNN). [7]

В компьютерном зрении большинство моделей сегментации изображений состоят из сети кодировщика-декодера. Кодер кодирует представление входных данных в скрытом пространстве, которое декодер декодирует для формирования карт сегментов или, другими словами, карт, описывающих местоположение каждого объекта на изображении.

Обычно процесс сегментации состоит из 3 этапов:

- 1) кодер изображений, который преобразует входное изображение в математическую модель (векторы и матрицы) для обработки;
- 2) кодер агрегирует векторы на нескольких уровнях;
- 3) быстрый декодер маски принимает внедренные изображения в качестве входных данных и создает маску, которая отдельно выделяет различные объекты на изображении.

Начиная с 2014 года появилась волна алгоритмов сегментации на основе глубокого обучения, таких как CNN+CRF [8] и FCN [6], которые добились значительного прогресса в этой области. В 2015 году наблюдался рост популярности U-Net [9] (рисунок 3) и сети деконволюции, что повысило точность результатов сегментации. На рисунке 3 U-Net от простого изображения головы слева до сегментированного результата справа, состоящего из: «воздуха» («air»), мягких тканей (soft tissue) и «костей» («bone»). [10] [11]

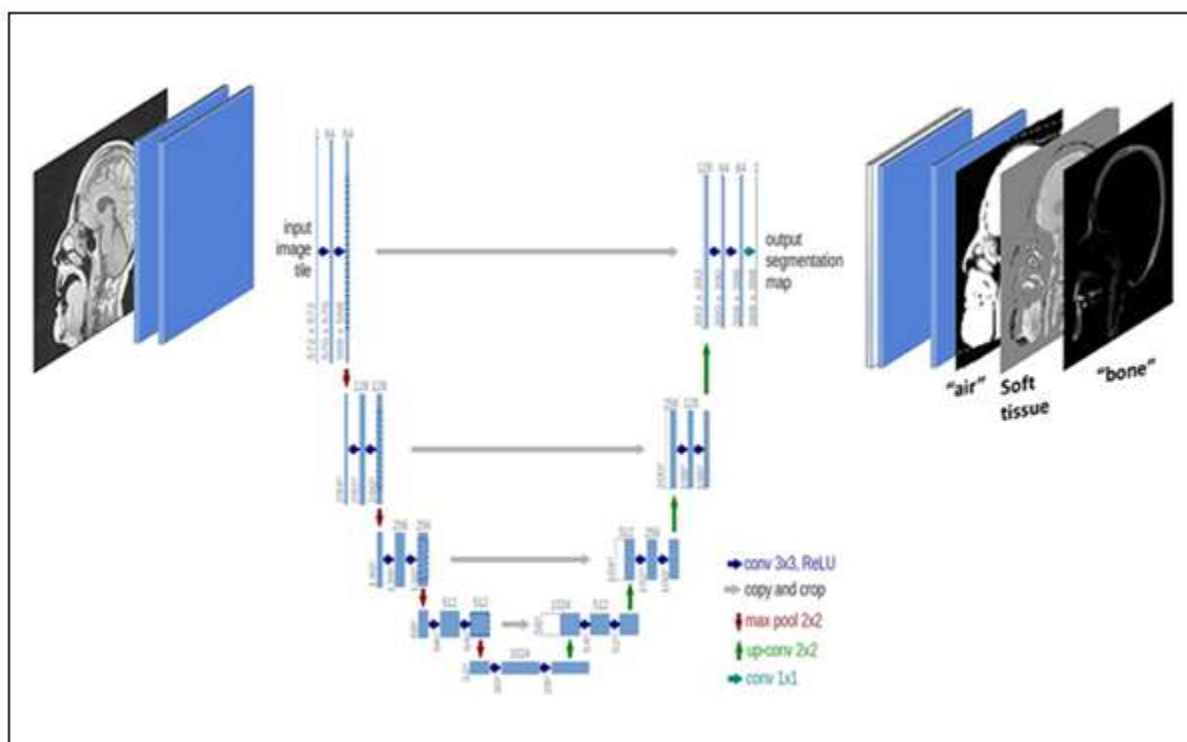


Рисунок 3 – Демонстрация структуры нейросети U-Net

Затем, в 2016 году, Instance Aware Segmentation [12], V-Net [13] и RefineNet [14] еще больше повысили точность и скорость сегментации. К 2017 году Mask-RCNN [15] и FC-DenseNet [16] представили обнаружение объектов и плотное прогнозирование для задач сегментации.

В 2018 году паноптическая сегментация, сети контекстного кодирования [17] были в центре внимания, поскольку эти подходы учитывали необходимость сегментации на уровне экземпляра. К 2019 году Panoptic FPN [18], HRNet [19] и Criss-Cross Attention [20] представили новые подходы к сегментации на уровне экземпляров.

В 2020 году эта тенденция продолжилась с появлением Panoptic DeepLab [21], PolarMask [22], CenterMask [23], DC-NAS [24] и Efficient Net [25] + NAS-FPN [26].

1.4 Задача классификации и обнаружения изображения

Классификация изображений (рисунок 4) является задачей компьютерного зрения, которая включает в себя присвоение классов цифровым изображениям. Это важный процесс в различных областях, включая медицину, безопасность и развлечения. На рисунке 4 показана классификация с помощью нейросети, где при помощи свертки матриц идет выбор между тем, что на выходе (Output) получено: кот (Cat) или собака (Dog). [27]

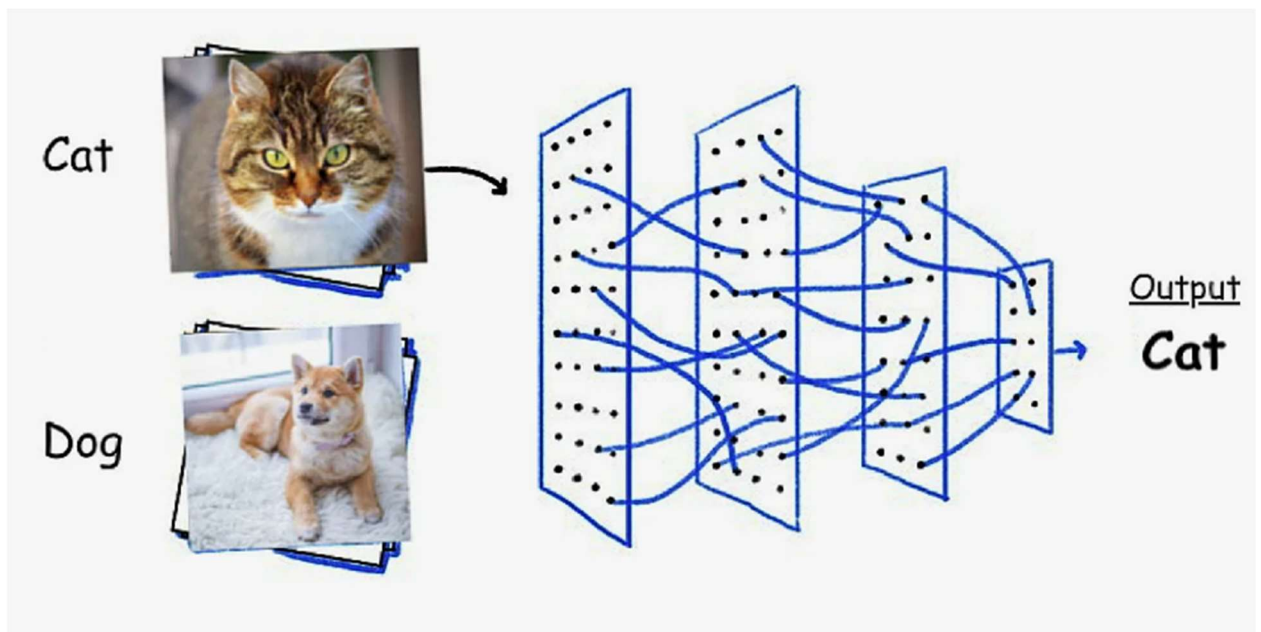


Рисунок 4 – Схематическая классификация изображения

Классификация изображений является процессом, который включает в себя категоризацию изображений по определенным классам. Задача классификации изображений включает в себя обучение компьютерной модели распознаванию различных закономерностей на изображениях. Модель учится на наборе помеченных изображений и применяет полученные знания для точной классификации новых изображений. Цель классификации изображений заключается в автоматизации анализа изображений, сделав его более эффективным и точным.

Для классификации изображений используется несколько методов. Для классификации изображений обычно используются традиционные алгоритмы машинного обучения, такие как машины опорных векторов (SVM) [28] и k-ближайшие соседи (k-NN) [29]. Эти алгоритмы основаны на созданных вручную функциях, которые извлекаются из изображений и используются для обучения моделей машинного обучения.

Однако алгоритмы глубокого обучения, такие как CNN (Convolution Neural Network), произвели революцию в классификации изображений. CNN предназначены для обработки данных изображений с минимальной предварительной обработкой, что делает их более эффективными и действенными, чем традиционные алгоритмы машинного обучения. CNN основаны на идее нейронной сети прямого распространения, где входными данными является изображение, а выходными данными — предсказанная метка или категория.

CNN является типом алгоритма глубокого обучения, который стал современным методом классификации изображений. CNN состоят из нескольких слоев, включая сверточные слои, слои пула и полносвязные слои.

Сверточные слои отвечают за изучение особенностей изображений. Сверточный слой применяет набор фильтров к входному изображению, создавая набор карт объектов, которые представляют различные функции изображения. Слои объединения используются для уменьшения размера карт объектов путем выполнения операции объединения, например максимального объединения или среднего объединения. Полносвязные слои отвечают за классификацию изображений на основе функций, полученных на сверточных слоях и слоях объединения.

Обучение CNN включает в себя подачу в сеть помеченных изображений и оптимизацию весов сети для минимизации ошибки классификации. Процесс оптимизации включает в себя расчет потерь между прогнозируемым выходным сигналом сети и истинными метками изображений. Затем веса сети обновляются с помощью обратного распространения ошибки, которое

включает в себя распространение ошибки обратно по уровням сети для корректировки весов.

Процесс обучения CNN требует большого набора данных помеченных изображений. Набор данных разделен на три набора: наборы для обучения, проверки и тестирования. Обучающий набор используется для обучения сети, набор проверки используется для мониторинга производительности сети во время обучения, а набор тестирования используется для оценки производительности сети после обучения.

Алгоритмы глубокого обучения, такие как CNN, произвели революцию в классификации изображений. Использование алгоритмов глубокого обучения при классификации изображений привело к значительному повышению точности и эффективности. Алгоритмы глубокого обучения использовались для различных задач классификации изображений, включая распознавание объектов, распознавание лиц и распознавание сцен.

Одно из самых популярных применений глубокого обучения для классификации изображений находится в области автономного вождения. Автономным транспортным средствам требуется способность распознавать объекты и точно их классифицировать, чтобы принимать решения о вождении. Алгоритмы глубокого обучения использовались для разработки систем, которые могут распознавать и классифицировать объекты, такие как автомобили, пешеходы и дорожные знаки.

Другое применение глубокого обучения в классификации изображений находится в области медицинской визуализации. Алгоритмы глубокого обучения использовались для разработки систем, которые могут обнаруживать и классифицировать заболевания на основе медицинских изображений, таких как рентгеновские снимки, МРТ и КТ. Эти системы могут совершить революцию в сфере здравоохранения, обеспечивая более быструю и точную диагностику.

Помимо повышения точности и эффективности, алгоритмы глубокого обучения также позволили разработать более сложные и совершенные

системы классификации изображений. Например, генеративно-сопоставительные сети (GAN) [30] использовались для создания реалистичных изображений и для передачи стилей, когда стиль одного изображения применяется к другому изображению. Эти приложения могут потенциально использоваться в таких областях, как искусство и дизайн.

Хотя алгоритмы глубокого обучения значительно повысили точность и эффективность классификации изображений, все еще существуют некоторые проблемы, которые необходимо решить. Одной из основных проблем является необходимость в больших наборах данных для обучения моделей глубокого обучения. Сбор и маркировка больших наборов данных может оказаться трудоемким и дорогостоящим процессом.

Еще одной проблемой является проблема предвзятости в классификации изображений. Модели глубокого обучения могут изучать предвзятости, присутствующие в обучающих данных, что может привести к предвзятой классификации новых данных. Этот вопрос становится все более важным, поскольку модели глубокого обучения используются в таких областях, как правоохранительная деятельность и здравоохранение, где предвзятая классификация может иметь серьезные последствия.

Классификация изображений является важным процессом в различных областях, и алгоритмы глубокого обучения, такие как CNN, значительно повысили его точность и эффективность. CNN состоят из нескольких слоев, в том числе сверточных слоев, слоев объединения и полностью связанных слоев, и обучаются с использованием больших наборов данных помеченных изображений.

Алгоритмы глубокого обучения также позволили разработать более сложные системы классификации изображений, такие как GAN, которые имеют потенциальное применение в таких областях, как искусство и дизайн. Однако по-прежнему существуют проблемы, которые необходимо решить, такие как необходимость в больших наборах данных и проблема систематической ошибки в классификации изображений.

В целом, достижения в области глубокого обучения и классификации изображений могут произвести революцию в различных отраслях и областях, и это интересная область исследований с множеством возможностей для дальнейшего развития и инноваций.

Есть много ситуаций, в которых полезно знать, где что-то находится на изображении или видео. Рассмотрим сценарий, в котором создается система спортивной аналитики. Если возможно определить, что находится в видеопотоке, показывающем игру, можно подсчитать игроков в регионе, отслеживать, как долго игроки находятся в разных регионах, и многое другое. Обнаружение объектов может открыть новые возможности для каждой отрасли и создать совершенно новые приложения, которые раньше были невозможны.

Именно здесь на помощь приходит обнаружение объектов. Обнаружение объектов (рисунок 5) является системой компьютерного зрения, которое определяет, что находится на изображении и где находятся обнаруженные объекты на изображении. С помощью обнаружения объектов можно также определить, где что-то находится на изображении. На рисунке 5 зелеными охватывающими прямоугольниками выделены объекты: человек (person), грузовик (truck), машина (car), рюкзак (backpack). [31]

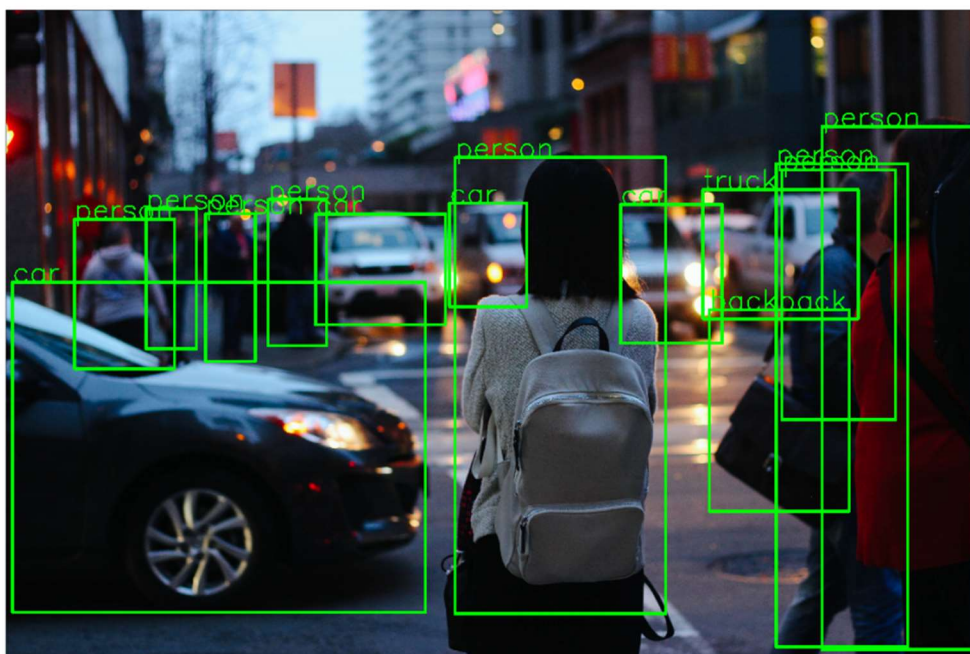


Рисунок 5 – Результат решения задачи классификации и обнаружения

Варианты использования обнаружения объектов весьма разнообразны: от выявления проблем с продуктами до поиска аномалий на медицинских изображениях, обнаружения трещин в рельсах и анализа видео для получения различной информации.

Система обнаружения объектов вернет координаты объектов на изображении, которые она была обучена распознавать. Система также вернет уровень уверенности, который показывает, насколько система уверена в точности прогноза.

На рисунке 4 все объекты выделены зелеными квадратами. Эти поля представляют собой координаты, в которых система обнаружения объектов, называемая моделью, считает, что объект присутствует. В индустрии компьютерного зрения мы называем рамки прогнозирования «ограничивающими рамками».

Как только становится понятно, что находится на изображении или видео, возможно применить постобработку, чтобы получить больше информации от визуальных медиа.

Модели обнаружения объектов необходимо обучать. Это относится к процессу, в ходе которого создается нейронная сеть и изучает особенности изображения. Нейронная сеть изучает особенности, показывая изображения объекта в различных сценариях (т. е. на разном фоне, объекты под разными углами) с метками, соответствующими объекту и его местоположению.

Изображения маркируются либо вручную, с использованием инструмента машинного зрения, такого как Roboflow Annotate, либо с использованием полуавтоматического решения, такого как Autodistill. Чем точнее и последовательнее аннотации и чем более репрезентативны изображения для среды, в которой будет развернута модель, тем лучше будет работать модель.

Эти функции изучаются и кодируются в «весах» и «предвзятости», которые сохраняются после обучения для использования при тестировании и производстве.

Обнаружение объектов можно использовать для решения любой проблемы, когда необходимо знать, содержит ли изображение или видео определенный объект или несколько объектов и где эти объекты расположены. Можно также использовать обнаружение объектов, когда нужно проверить, присутствует ли что-то на изображении или видео.

Область обнаружения объектов не так нова, как может показаться. Фактически, обнаружение объектов развивалось за последние 20 лет. Прогресс в обнаружении объектов обычно разделяется на два отдельных исторических периода (до и после внедрения глубокого обучения):

Детекция объектов до 2014 г. является периодом классического обнаружения объектов:

1) Детектор Виолы-Джонса [32] (2001 г.), работа, положившая начало развитию традиционных методов обнаружения объектов. Методы обнаружения объектов;

2) HOG Detector [33] (2006 г.), популярный дескриптор функции обнаружения объектов в компьютерном зрении и обработке изображений;

3) DPM [34] (2008 г.) с первым введением регрессии ограничивающего прямоугольника.

Детектор объектов после 2014 года является периодом обнаружения глубокого обучения

Важнейшие двухэтапные алгоритмы обнаружения объектов:

- 1) RCNN [35] и SPPNet [36] (2014 г.);
- 2) Fast RCNN [37] и Faster RCNN [38] (2015);
- 3) Mask R-CNN [39] (2017);
- 4) Пирамидальные сети/FPN [40] (2017);
- 5) G-RCNN [41] (2021).

Важнейшие одноэтапные алгоритмы обнаружения объектов:

- 1) YOLO [42] (2016);
- 2) SSD [43] (2016 г.);
- 3) RetinaNet [44] (2017);
- 4) YOLOv3 [45] (2018);
- 5) YOLOv4 [46] (2020);
- 6) YOLOR [47] (2021);
- 7) YOLOv7 [48] (2022);
- 8) YOLOv8 [49] (2023).

Чтобы понять, какой алгоритм лучше всего подходит для конкретного случая использования, важно понять его основные характеристики. Прежде чем обсуждать отдельные алгоритмы, будут рассмотрены ключевые различия между соответствующими алгоритмами распознавания изображений для обнаружения объектов.

Как видно из приведенного выше списка, современные методы обнаружения объектов можно разделить на два основных типа: одноэтапные и двухэтапные детекторы объектов.

Детекторы объектов на основе глубокого обучения извлекают признаки из входного изображения или видеокadra. Детектор объектов решает две

последующие задачи. Первая задача заключается в нахождении произвольного количества объектов (возможно, даже ноль), и вторая задача заключается в классификации каждого отдельного объекта и оценки его размера с помощью ограничивающей рамки.

Другие методы объединяют обе задачи в один этап (одноступенчатые детекторы [50]) для достижения более высокой производительности за счет точности.

В двухэтапных детекторах [51] объектов приблизительные области объекта предлагаются с использованием глубоких особенностей, прежде чем эти особенности будут использоваться для классификации изображений и регрессии ограничивающего прямоугольника для объекта-кандидата.

1) двухэтапная архитектура включает в себя предложение области объекта с использованием традиционных методов компьютерного зрения или глубоких сетей, за которым следует классификация объектов на основе функций, извлеченных из предлагаемой области, с помощью регрессии ограничивающего прямоугольника;

2) двухэтапные методы обеспечивают высочайшую точность обнаружения, но обычно работают медленнее. Из-за большого количества шагов вывода на изображение производительность (количество кадров в секунду) не так хороша, как у одноэтапных детекторов;

3) различные двухэтапные детекторы включают сверточную нейронную сеть региона (RCNN) с эволюцией Faster R-CNN или Mask R-CNN. Последней эволюцией является гранулированный RCNN (G-RCNN);

4) двухэтапные детекторы объектов сначала находят интересующую область и используют эту обрезанную область для классификации. Однако такие многоступенчатые детекторы обычно не поддаются сквозному обучению, поскольку обрезка является недифференцируемой операцией.

Одноэтапные детекторы. Одноэтапные детекторы прогнозируют ограничивающие рамки над изображениями без шага предложения области.

Этот процесс занимает меньше времени и поэтому может использоваться в приложениях реального времени.

1) одноэтапные детекторы объектов отдают приоритет скорости вывода и работают очень быстро, но не так хорошо распознают объекты неправильной формы или группу небольших объектов;

2) к самым популярным одноступенчатым детекторам относятся YOLO, SSD и RetinaNet. Новейшими детекторами реального времени являются YOLOv7 (2022 г.), YOLOR (2021 г.) и YOLOv4-Scaled (2020 г.).

3) основные преимущества обнаружения объектов с помощью одноэтапных алгоритмов включают, как правило, более высокую скорость обнаружения, а также большую структурную простоту и эффективность по сравнению с многоступенчатыми детекторами.

1.5 Окончательный выбор нейросети

Так как для задачи дипломного проекта требуется именно обнаружения объектов как можно более быстрым и эффективным методом, в таком случае, выбор пал на RetinaNet [44], SSD [43], или YOLO [42] нейросетевые решения, так как они являются самыми быстродействующими из существующими на сегодняшний день одноэтапными детекторами.

RetinaNet конвейер обнаружения слишком сложен для понимания, поддержки и развертывания. Для полного обеспечения обнаружения потребуется ручная настройка большого числа гиперпараметров, такие как: генератор «якорей», функция ошибки, функция отзыва, TensorFlow конвейер, функции параметров охватывающих прямоугольников. [52] Даже полная настройка всей этой системы не гарантирует результат для реальных данных.

Нейросетевое решение SSD (Single Shot [MultiBox] Detector) ориентировано на операционную систему Linux, это полностью открытое программное обеспечение. Проблема в том, что официальная реализация этой

нейросети не имеет подробного tutorиала её обучения, потому что SSD как таковая является научной работой, а не коммерческим продуктом, готовым к внедрению.

Самой подробно описанной, популярной и простой в освоении системой обнаружения объектов является YOLO, на которую и пал выбор. У YOLO версии 8 даже есть собственная библиотека Python – ultralytics. [53]

YOLO (You Only Look Once) стал прорывом в области обнаружения объектов, поскольку это был первый одноэтапный подход к обнаружению объектов, который рассматривал обнаружение как проблему регрессии. Архитектура обнаружения просматривала изображение только один раз, чтобы предсказать расположение объектов и метки их классов.

В отличие от подхода с двухэтапным детектором (Fast RCNN, Faster RCNN), YOLOv1 [54] не имеет генератора предложений и этапов уточнения; он использует одну нейронную сеть, которая прогнозирует вероятности классов и координаты ограничивающего прямоугольника на основе всего изображения за один проход. Его можно оптимизировать сквозным образом, поскольку конвейер обнаружения представляет собой одну сеть; думайте об этом как о сети классификации изображений.

Поскольку сеть предназначена для сквозного обучения, аналогичного классификации изображений, ее архитектура чрезвычайно быстра, а базовая модель YOLO прогнозирует изображения со скоростью 45 кадров в секунду (кадров в секунду), измеренную на графическом процессоре Titan X. Авторы также придумали гораздо более легкую версию YOLO под названием Fast YOLO, в которой меньше слоев, обрабатывающих изображения со скоростью 155 кадров в секунду.

YOLO достиг 63,4 мА (средняя средняя точность), что более чем вдвое превышает показатели других детекторов реального времени, что делает его еще более особенным. YOLO, и Fast YOLO превосходят варианты DPM для обнаружения объектов в реальном времени со значительным отрывом с точки зрения средней точности (почти в 2 раза) и FPS.

Редмон и Фархади (2017) опубликовали статью YOLO9000 [55]: Better, Faster, Stronger на конференции CVPR. В этой статье авторы предложили два современных варианта YOLO: YOLOv2 [56] и YOLO9000; оба были идентичны, но различались стратегией обучения.

YOLOv2 был обучен на таких наборах данных обнаружения, как Pascal VOC и MS COCO. В то же время YOLO9000 был разработан для прогнозирования более 9000 различных категорий объектов путем совместного обучения на наборах данных MS COCO и ImageNet.

В улучшенной модели YOLOv2 использовались различные новые методы, позволяющие превзойти современные методы, такие как Faster-RCNN и SSD, как по скорости, так и по точности. Одним из таких методов было многомасштабное обучение, которое позволяло сети прогнозировать при различных размерах входных данных, обеспечивая тем самым компромисс между скоростью и точностью.

Редмон и Фархади (2018) опубликовали документ YOLOv3 [45]: постепенное улучшение arXiv. Авторы внесли множество изменений в архитектуру сети и адаптировали большинство других методов YOLOv1 и особенно YOLOv2.

В этом документе представлена новая сетевая архитектура под названием Darknet-53 [57]. Darknet-53 — гораздо более крупная сеть, чем раньше, она намного точнее и быстрее. Он обучается на различных разрешениях изображений, таких как 320×320 , 416×416 . При разрешении 320×320 YOLOv3 [45] достигает 28,2 mAP при частоте 45 кадров в секунду на графическом процессоре Titan X и работает так же точно, как детектор одиночных снимков (SSD321), но в 3 раза быстрее.

YOLOv4 [46] — это продукт множества экспериментов и исследований, в которых сочетаются различные небольшие новые методы, улучшающие точность и скорость сверточной нейронной сети.

В этой статье были проведены обширные эксперименты с различными архитектурами графических процессоров и показано, что YOLOv4

превосходит все другие архитектуры сетей обнаружения объектов с точки зрения скорости и точности.

В 2020 г. Бочковский и др. (автор известного репозитория GitHub: Darknet) опубликовал статью YOLOv4: Оптимальная скорость и точность обнаружения объектов на arXiv [58]. YOLOv4 работает в два раза быстрее, чем EfficientDet, при сопоставимой производительности, а также улучшает mAP и FPS YOLOv3 на 10% и 12%.

В 2020 году, после выпуска YOLOv4, всего через два месяца, Гленн Джохер, основатель и генеральный директор Ultralytics, опубликовал на GitHub реализацию YOLOv5 с открытым исходным кодом. YOLOv5 предлагает семейство архитектур обнаружения объектов, предварительно обученных на наборе данных MS COCO. За ним последовал выпуск EfficientDet и YOLOv4. Это единственный детектор объектов YOLO, у которого нет исследовательской статьи, которая изначально вызвала некоторые споры.

Сегодня YOLOv5 [59] является одной из официальных современных моделей с огромной поддержкой, которую проще использовать в производстве. Самое приятное то, что YOLOv5 изначально реализован в PyTorch, устраняя ограничения платформы Darknet (основанной на языке программирования C и не предназначенной для производственной среды). Платформа Darknet развивалась с течением времени и представляет собой отличную исследовательскую среду для работы, обучения, тонкой настройки и вывода с помощью TensorRT; все это возможно с Darknet. Однако у него меньшее сообщество и, следовательно, меньшая поддержка.

Это огромное изменение YOLO в PyTorch облегчило разработчикам модификацию архитектуры и прямой экспорт во многие среды развертывания. YOLOv5 является одной из официальных современных моделей, представленных на витрине Torch Hub.

Meituan YOLOv6 [60] является современным детектором объектов, который предлагает замечательный баланс между скоростью и точностью, что

делает его популярным выбором для приложений реального времени. В этой модели реализовано несколько заметных улучшений в ее архитектуре и схеме обучения, включая реализацию модуля двунаправленной конкатенации (BiC), стратегии обучения с привязкой (AAT), а также улучшенную конструкцию магистрали и «шей» нейросети.

YOLOv7 [48] является современным детектором объектов в реальном времени, превосходящий все известные детекторы объектов как по скорости, так и по точности в диапазоне от 5 до 160 кадров в секунду. Он имеет самую высокую точность (56,8% AP) среди всех известных детекторов объектов реального времени с частотой 30 кадров в секунду и выше на графическом процессоре V100. Более того, YOLOv7 превосходит другие детекторы объектов, такие как YOLOR, YOLOX, Scaled-YOLOv4, YOLOv5 и многие другие, по скорости и точности.

YOLOv8 [49] является новейшей версией серии детекторов объектов в реальном времени YOLO, предлагающая передовые характеристики с точки зрения точности и скорости. Основываясь на достижениях предыдущих версий YOLO, YOLOv8 представляет новые функции и оптимизации, которые делают его идеальным выбором для различных задач обнаружения объектов в широком спектре приложений.

Ключевая особенности YoloV8:

1) YOLOv8 использует самые современные архитектуры магистрали и «шей», что приводит к улучшению извлечения особенностей и производительности обнаружения объектов;

2) YOLOv8 использует разделенную голову Ultralytics без якоря, которая способствует повышению точности и более эффективному процессу обнаружения по сравнению с подходами на основе якоря;

3) YOLOv8 ориентирован на поддержание оптимального баланса между точностью и скоростью и подходит для задач обнаружения объектов в реальном времени в различных областях применения;

4) Разнообразие предварительно обученных моделей: YOLOv8 предлагает ряд предварительно обученных моделей для удовлетворения различных задач и требований к производительности, что упрощает поиск подходящей модели для вашего конкретного случая использования.

Метрики YOLOv8 играют ключевую роль в оценке эффективности моделей обнаружения объектов. Они обеспечивают количественную оценку того, насколько хорошо модель справляется с конкретными задачами. YOLOv8 использует набор показателей для оценки своей производительности, каждый из которых служит уникальной цели при оценке различных аспектов возможностей модели.

MAP (медиана средней точности), mAP — это широко используемый показатель при обнаружении объектов, который сочетает в себе точность и полноту для оценки точности модели. Он вычисляет среднюю точность для каждого класса, а затем вычисляет среднее значение для всех классов. YOLOv8 использует mAP для общей оценки точности модели при обнаружении объектов различных категорий.

Точность измеряет количество положительных прогнозов и является отношением истинно положительных прогнозов к сумме истинно положительных прогнозов и неверно положительных прогнозов. Полнота (recall), с другой стороны, оценивается способность модели фиксировать все релевантные случаи распознавания, представляя отношение истинно положительных прогнозов к сумме истинно положительных прогнозов и неверно отрицательных прогнозов.

YOLOv8 использует показатели точности и полноты для оценки компромисса между точностью и полнотой.

Оценка F1 представляет собой среднее гармоническое значение точности и полноты. Он обеспечивает сбалансированную меру, учитывающую как ложноположительные, так и ложноотрицательные результаты. YOLOv8 использует оценку F1 для оценки общей производительности модели, особенно когда точность и полнота имеют одинаковое значение.

YOLOv8 уделяет большое внимание обнаружению объектов в реальном времени. Поэтому показатели скорости, такие как количество кадров в секунду (FPS) и время вывода, имеют решающее значение для оценки его эффективности. Эти показатели оценивают, насколько быстро YOLOv8 может обрабатывать и анализировать изображения, что делает его подходящим для приложений, где важна низкая задержка.

2 Проектирование метода

Работа с нейросетями в области обнаружения объектов на термовизионных аэро-фотоснимках означает наличие нескольких шагов в процессе формирования весов:

- 1) загрузка набора данных в инструмент аннотации изображений;
- 2) разметка данных;
- 3) выгрузка данных в нужном для нейросети формате;
- 4) подготовка набора данных для обучения нейросети;
- 5) процесс обучения нейросети на наборе данных;
- 6) использование полученных в ходе обучения нейросети весов.

Суть метода заключается в зацикливании этого процесса для сокращения работы по аннотации изображений. Зачастую для обучения требуются тысячи изображений, которые нужно разметить полностью вручную. Использование же полученных на шаге 6 весов нейросети позволяет сократить время аннотации набора данных, достаточно лишь проверить качество аннотаций нужных объектов, а не выделять все объекты вручную.

В таком случае, получается следующий алгоритм метода, изображенный на рисунках 6 и 7. На рисунке 6 на этапе «выгрузка размеченного набора данных» происходит начало зацикливания процесса получения более совершенных весов, на данной UML диаграмме сделан акцент на передачу разметки и набора данных.

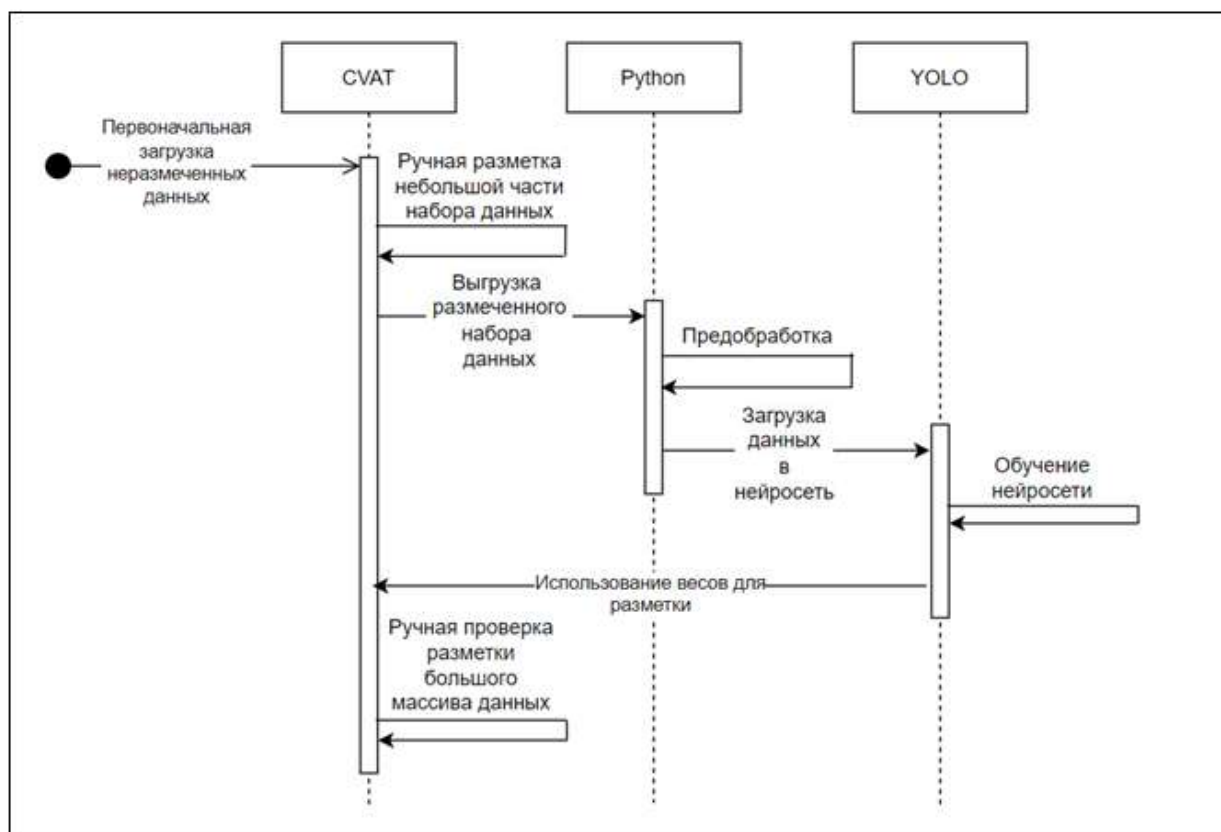


Рисунок 6 – Схематическое изображение алгоритма метода

Работа с весами нейросети в получившейся системе изображена на рисунке 7, где наглядно показана передача весов при повторных итерациях метода обнаружения объектов. Веса, полученные в ходе обучения нейросети, используются вместе с данными, чтобы получить ещё более совершенные веса.

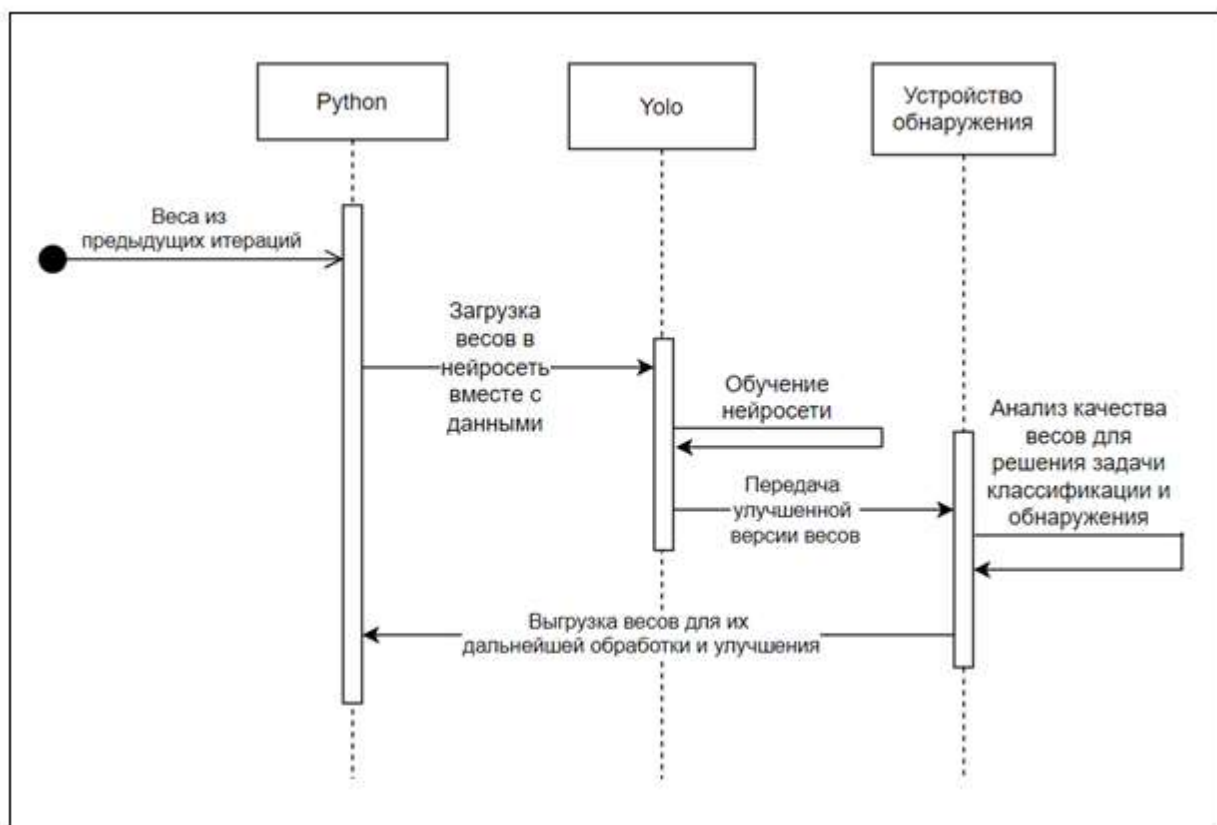


Рисунок 7 – Диаграмма передачи весов

Все начинается с ручной разметки сырых данных, потом небольшая часть набора данных выгружается и подготавливается к обучению на ней нейросети. Нейросеть обучается, откуда получают веса нейросети, которые используются для дополнительной аннотации большого числа изображений набора данных. Полученная разметка вручную проверяется, что легче, чем вручную размечать полностью весь набор данных, после чего размеченный набор данных выгружается, и с этого момента процесс обучения нейросети закидывается до получения самого лучшего набора весов для наилучшего обнаружения объектов на термовизионных аэро-фотоснимках.

3 Работа с предоставленными данными

Были предоставлены видео с целью обнаружения на них объектов, однако, мне нельзя их показывать, поэтому для демонстрации непосредственно обнаружения, без метрик, будет использован открытый набор данных из интернета HIT-UAV. [61] Метрики будут показаны, основанные на большом объеме реальных данных.

Было запущено web-приложение CVAT с помощью Docker [62] на базе WSL. [63] Далее было размечено 200 кадров из разных частей видео, чтобы можно было подчерпнуть максимальное количество полезной информации из предоставленных данных. Ещё 200 кадров фона было добавлено в датасет помимо 200 размеченных кадров для большего разнообразия данных. Этот небольшой кусочек набора данных был, далее обучен, и получен первоначальный результат для автоматической частичной разметки набора данных с помощью нейросети.

Такое маленькое количество кадров приводит к появлению ряда проблем при последующей разметке набора данных. Например, на рисунке 8 изображено выделение нейросетью несколько раз одного и того же, но с разными охватывающими прямоугольниками.

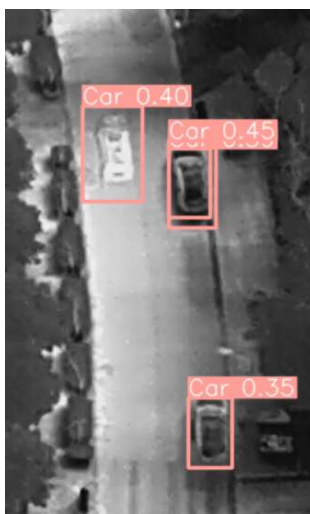


Рисунок 8 – Ошибка обнаружения

Также может быть выделено совсем не то, что подразумевается, такие аннотации нужно удалить в ходе повторной разметки набора данных. На рисунке 9 видна ошибка классификации, прожектор на стадионе выделен как машина (Car).



Рисунок 9 – Ошибка классификации

Ещё может быть такая проблема, что нейросеть выделяет частично не то, что нужно, такое тоже необходимо править в ходе повторной аннотации. На рисунке 10 машина снизу выделена лишь частично вместе с землей, также видно, что некоторые машины не выделены вовсе.



Рисунок 10 – Ошибка обнаружения

После первой итерации мною было размечено 4917 изображений реального набора данных, и эти изображения были использованы в ходе обучения нейронной сети. Были использованы 3687 изображений непосредственно для обучения нейросети, и 1230 изображений для валидации во время обучения, чего вполне достаточно для решения задачи обнаружения нужных объектов.

Далее будут описаны метрики, которые получились в ходе обучения такого большого объема данных. Во-первых, на рисунке 11 изображена матрица неточностей распознавания. В ней слева-направо сверху-вниз изображены обнаружения: истинно положительные, неверно положительные, неверно отрицательные и истинно отрицательные. Истинно отрицательных нет, потому что объект для обнаружения только один. На оси ординат находится истинное значение обнаружения, состоящее из: обнаруживаемого объекта и фона. На оси абсцисс находится полученный результат предсказания, состоящий тоже из обнаруживаемого объекта и фона.



Рисунок 11 – Матрица ошибок распознавания

Коррелограмма разметки – это термин, используемый в области машинного обучения, который относится к графику, используемому для отображения корреляции между метками, предсказанными алгоритмом обнаружения объектов во время обучения.

В частности, коррелограмма разметки – это график цветовой матрицы, который показывает корреляцию между метками данных обучающего набора. Это может помочь понять поведение и производительность алгоритмов обнаружения объектов во время обучения, а также взаимодействие между прогнозируемыми метками.

Наблюдая за коррелограммой разметки, можно понять способность алгоритма обнаружения целей различать разные категории, а также точность прогнозирования для разных категорий. Кроме того, также можно оценить производительность алгоритма и качество набора данных, сравнивая коррелограммы разметки разных алгоритмов или разных наборов данных.

Коррелограмма разметки большого набора реальных данных изображена на рисунке 12, где на оси ординат и абсцисс расположены параметры меток, их: положение x , положение y , ширина, высота. Наибольший интерес представляют два графика, выделенные зелеными квадратами, их значение пояснено далее.

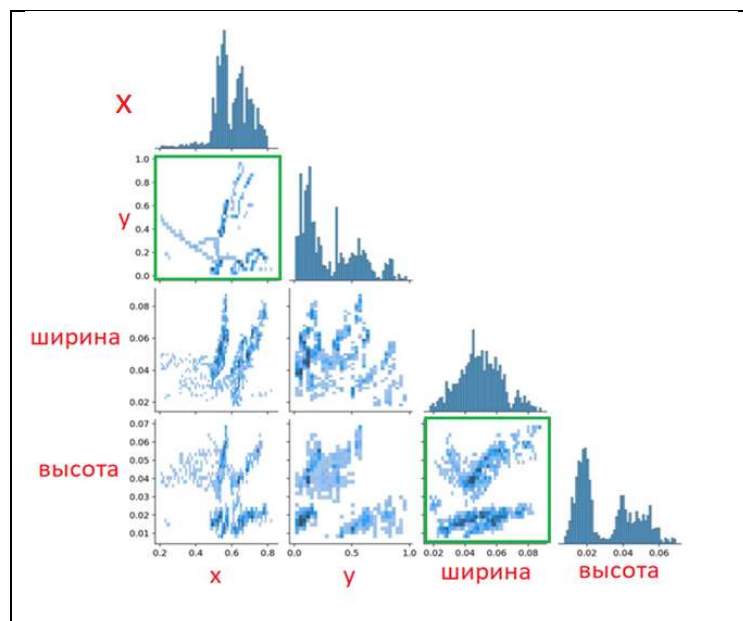


Рисунок 12 – Коррелограмма разметки

Далее идёт график меток по ширине-высоте и расположения на кадрах, рисунок 13, где те самые «представляющие интерес» графики из предыдущего рисунка 11. Оси: координаты x , y , высота (height), ширина (width). График слева показывает, на каких координатах обычно встречаются на изображениях метки, а справа тепловая карта показывает, в каких пределах ранжируются метки по относительной совокупной высоте и ширине.

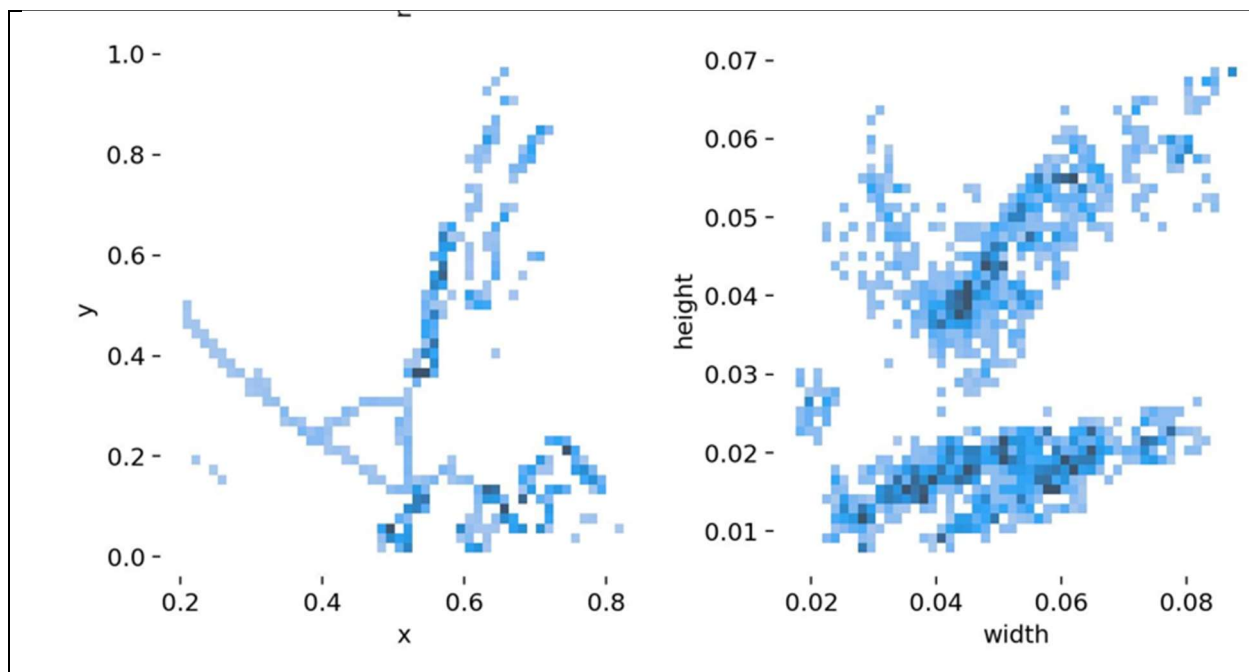


Рисунок 13 – Графики характеристик меток

Далее график PR-кривой (Precision-Recall), он показывает взаимосвязь между точностью и полнотой модели при различных порогах классификации.

Чем ближе кривая PR находится к верхнему левому углу оси координат, тем лучше производительность модели, тем точнее она может идентифицировать положительные образцы и тем выше значение точности правильной классификации положительных образцов. Если он ближе вправо, это означает, что модель имеет меньшую способность идентифицировать положительные образцы, то есть способность запоминания плохая.

Характеристика кривой PR заключается в том, что при изменении порога классификации точность и скорость полноты будут соответственно меняться. Обычно кривая PR находится в более высоком положении, когда модель классификации способна поддерживать как высокую точность, так и высокую полноту. Когда модель смещена в сторону высокой точности или высокой полноты, кривая соответственно будет двигаться в направлении низкой точности или низкой полноты.

Кривая PR может помочь нам оценить производительность модели при различных пороговых значениях и выбрать соответствующий порог, чтобы сбалансировать точность и полноту. Для сравнения или выбора модели мы можем провести количественную оценку, сравнивая площадь под кривой PR (так называемая средняя точность, AP, Average Precision). Чем больше значение AP, тем лучше производительность модели.

Для реального набора данных PR кривая показана на рисунке 14, где на оси ординат расположена полнота, а на оси абсцисс расположена точность. Как видно из числа справа-сверху 0.947, достигнута высокая точность при высокой полноте обнаружения объектов, что говорит о хорошем качестве конечного результата.

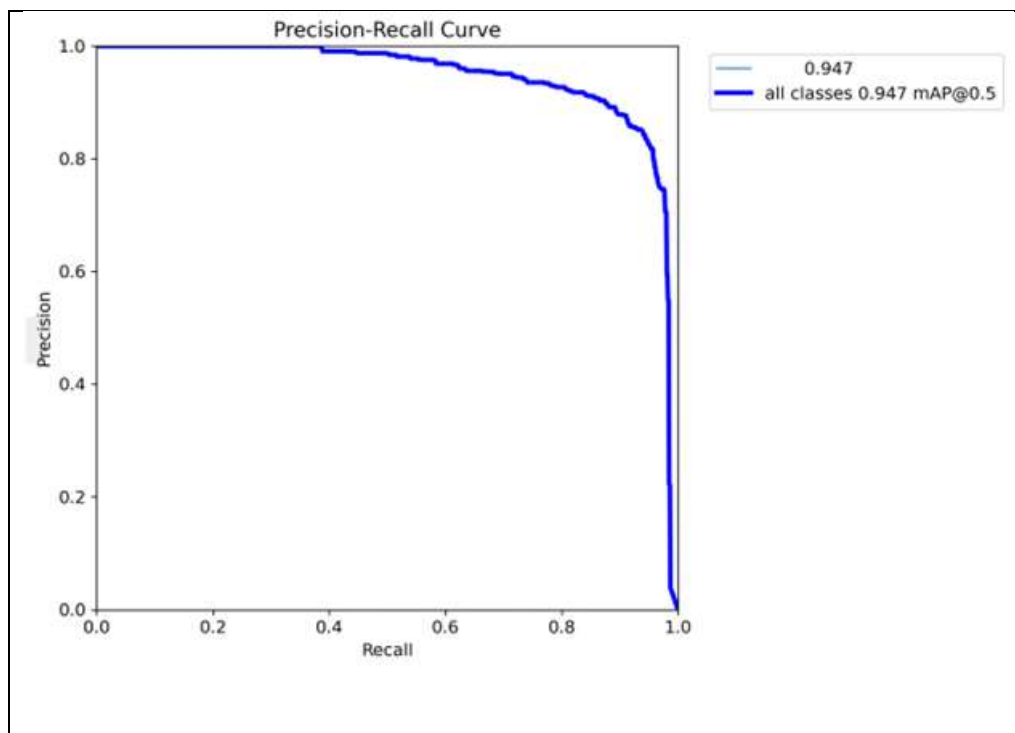


Рисунок 14 – PR-кривая

Таким образом, кривая PR – это визуальный инструмент, который показывает взаимосвязь между точностью и полнотой модели классификации. Нарисовав кривую точности-отзыва, мы можем оценить и сравнить производительность модели при различных порогах классификации и рассчитать среднюю точность. Среднее значение (AP) используется для количественной оценки качества модели.

ЗАКЛЮЧЕНИЕ

В ходе выпускной квалификационной работы были выполнены следующие задачи:

- проведен анализ первоначальных требований к методу и предметной области, где было сформировано понимание верхнеуровневых требований и функциональных возможностей систем обнаружения объектов на термовизионных аэро-фотоснимках;

- проведен сравнительный анализ методов для обнаружения объектов с помощью изучения различных решений в сети интернет и непосредственной работы с этими методами, благодаря чему были выявлены ключевые преимущества решений. Многие из которых были занесены в список возможного для реализации функционала, а также выявлены недостатки готовых решений на рынке, что легло в основу проектирования метода;

- собраны требования от научного руководителя;

- произведен анализ требований к системе. Совместно с научным руководителем проекта определены приоритеты задач на реализацию и границы проекта, а также сформированы функциональные и нефункциональные требования.

По итогам работы в соответствии с методическими указаниями написана пояснительная записка к выпускной квалификационной работе. [64]

На основе достигнутых результатов выполненной работы проект метода для обнаружения объектов на термовизионных аэро-фотоснимках:

- код выложен в открытый доступ на ресурс GitHub с демонстрационной целью; [65]

- благодаря системе версий Git, проект можно разрабатывать, используя итерации кода, что позволяет сохранять предыдущие решения при разработки новых деталей метода.

Рассмотренные в рамках выпускной квалификационной работы вопросы являются немаловажными при разработке подобных методов. Благодаря уже

завершенным и описанным этапам метода обнаружения объектов на аэро-фотоснимках можно лучше распределить имеющиеся ресурсы команды разработки и развивать свой метод для удовлетворения потребностей заказчика, которые не вошли в первую версию продукта.

Дипломный проект выполнен в интересах УПКБ Деталь и Уральского Отделения Российской Академии Наук.

Выпускная квалификационная работа закончена «2» мая 2024 г.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Определение аффинного преобразования в Большой Советской Энциклопедии. – URL: http://gatchina3000.ru/big/083/035_bolshaya-sovetskaya.htm (дата обращения 30.04.2024).
2. CVAT: Computer Vision Annotation Tool – 2024 Guide. – URL: <https://viso.ai/computer-vision/cvat-computer-vision-annotation-tool/> (дата обращения 30.04.2024).
3. Segment Anything Model – Computer Vision Gets A Massive Boost. – URL: <https://www.unite.ai/segment-everything-model-computer-vision-gets-a-massive-boost/> (дата обращения 30.04.2024).
4. Что такое сегментация изображения. – URL: <https://triptonkosti.ru/4-foto/chto-takoe-segmentaciya-izobrazhenij-94-foto.html> (дата обращения 30.04.2024).
5. Как свёрточные нейросети имитируют работу мозга. – URL: <https://practicum.yandex.ru/blog/svertochnye-neyronnye-seti> (дата обращения 30.04.2024).
6. Understanding Fully Convolutional Networks. – URL: <https://jianchao-li.github.io/post/understanding-fully-convolutional-networks/> (дата обращения 30.04.2024).
7. Recurrent Neural Network. – URL: <https://kobiso.github.io/research/research-rnn/> (дата обращения 30.04.2024).
8. CRF Learning with CNN Features for Image Segmentation. – URL: <https://arxiv.org/pdf/1503.08263> (дата обращения 30.04.2024).
9. Официальный Гитхаб U-Net. – URL: <https://github.com/hoangsaobao/U-net> (дата обращения 30.04.2024).
10. U NETWORK. – URL: <https://flectone.ru/u-network.html> (дата обращения 30.04.2024).

11. Изображение U-Net из официального репозитория. – URL: <https://github.com/hoangcaobao/U-net/blob/main/image/u-net-architecture.png> (дата обращения 30.04.2024).
12. Instance-aware Semantic Segmentation via Multi-task Network Cascades. – URL: <https://arxiv.org/pdf/1512.04412> (дата обращения 30.04.2024).
13. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. – URL: <https://arxiv.org/pdf/1606.04797> (дата обращения 30.04.2024).
14. RefineNet: Enhancing Text-to-Image Conversion with High Resolution and Detail Accuracy through Hierarchical Transformers and Progressive Refinement. – URL: <https://arxiv.org/pdf/2312.17274> (дата обращения 30.04.2024).
15. Официальный Гитхаб Mask-RCNN. – URL: https://github.com/matterport/Mask_RCNN (дата обращения 30.04.2024).
16. Официальный Гитхаб FC-DenseNet. – URL: <https://github.com/SimJeg/FC-DenseNet> (дата обращения 30.04.2024).
17. Context Encoding for Semantic Segmentation. – URL: https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_Context_Encoding_for_CVPR_2018_paper.pdf (дата обращения 30.04.2024).
18. Panoptic Feature Pyramid Networks. – URL: <https://arxiv.org/abs/1901.02446> (дата обращения 30.04.2024).
19. Официальный Гитхаб HRNet. – URL: <https://github.com/HRNet> (дата обращения 30.04.2024).
20. CCNet: Criss-Cross Attention for Semantic Segmentation. – URL: <https://arxiv.org/pdf/1811.11721v1> (дата обращения 30.04.2024).
21. Официальный Гитхаб Panoptic Deeplab. – URL: <https://github.com/bowenc0221/panoptic-deeplab> (дата обращения 30.04.2024).
22. Официальный Гитхаб PolarMask. – URL: <https://github.com/xieenze/PolarMask> (дата обращения 30.04.2024).

23. Официальный Гитхаб CenterMask. – URL: <https://github.com/youngwanLEE/CenterMask> (дата обращения 30.04.2024).
24. DC-NAS: Divide-and-Conquer Neural Architecture Search. – URL: <https://arxiv.org/abs/2005.14456> (дата обращения 30.04.2024).
25. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. – URL: <https://arxiv.org/abs/1905.11946> (дата обращения 30.04.2024).
26. Официальный Гитхаб NAS-FPN. – URL: https://github.com/DetectionTeamUCAS/NAS_FPN_Tensorflow (дата обращения 30.04.2024).
27. Evaluate the Classification Performance of Machine Learning Models using TensorFlow: A Computer Vision Example. – URL : <https://www.codementor.io/@umairchoudhry/evaluate-the-classification-performance-of-machine-learning-models-using-tensorflow-a-computer-vision-example-1ad5cdoi7u> (дата обращения 30.04.2024).
28. Support vector machines. – URL: <https://scikit-learn.org/stable/modules/svm.html> (дата обращения 30.04.2024).
29. OpenSearch k-NN. – URL: <https://github.com/opensearch-project/k-NN> (дата обращения 30.04.2024).
30. Generative Adversarial Networks. – URL: <https://arxiv.org/abs/1406.2661> (дата обращения 30.04.2024).
31. Распознавание изображений и обнаружение объектов. – URL: <https://waksoft.susu.ru/2021/05/27/image-recognition-and-object-detection-part1/> (дата обращения 30.04.2024).
32. Viola and Jones Object Detection Framework (C++). – URL: <https://github.com/alexdemartos/ViolaAndJones> (дата обращения 30.04.2024).
33. Histogram of Oriented Gradients and Object Detection. – URL: <https://github.com/chriamue/hog-detector> (дата обращения 30.04.2024).
34. Deformable Part Models (DPMs) for Human Detection. – URL: https://fudanimc.github.io/slides/dpmv2_zgz.pdf (дата обращения 30.04.2024).
35. Официальный Гитхаб RCNN. – URL : <https://github.com/rbgirshick/rcnn> (дата обращения 30.04.2024).

36. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. – URL: <https://arxiv.org/abs/1406.4729> (дата обращения 30.04.2024).

37. Fast R-CNN: Fast Region-based Convolutional Networks for object detection. – URL: <https://github.com/rbgirshick/fast-rcnn> (дата обращения 30.04.2024).

38. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. – URL: <https://github.com/rbgirshick/py-faster-rcnn> (дата обращения 30.04.2024).

39. Mask R-CNN. – URL: <https://arxiv.org/abs/1703.06870> (дата обращения 30.04.2024).

40. Feature Pyramid Networks for Object Detection. – URL: <https://arxiv.org/abs/1612.03144> (дата обращения 30.04.2024).

41. GRCNN: Graph Recognition Convolutional Neural Network for Synthesizing Programs from Flow Charts. – URL: <https://arxiv.org/abs/2011.05980> (дата обращения 30.04.2024).

42. You Only Look Once: Unified, Real-Time Object Detection. – URL: <https://arxiv.org/abs/1506.02640> (дата обращения 30.04.2024).

43. SSD: Single Shot MultiBox Detector. – URL: <https://arxiv.org/pdf/1512.02325v4> (дата обращения 30.04.2024).

44. RetinaNet Examples. – URL: <https://github.com/florischabert/retinanet> (дата обращения 30.04.2024).

45. Официальный Гитхаб YoloV3. – URL : <https://github.com/ultralytics/yolov3> (дата обращения 30.04.2024).

46. Yolo v4, v3 and v2 for Windows and Linux. – URL: https://github.com/kiyoshiiriemon/yolov4_darknet (дата обращения 30.04.2024).

47. YOLOR. – URL: <https://github.com/WongKinYiu/yolor> (дата обращения 30.04.2024).

48. Official YOLOv7. – URL: <https://github.com/WongKinYiu/yolov7> (дата обращения 30.04.2024).

49. YOLOv8 Docs. – URL: <https://docs.ultralytics.com/models/yolov8/> (дата обращения 30.04.2024).
50. One-Stage Object Detection Models. – URL: <https://paperswithcode.com/methods/category/one-stage-object-detection-models/> (дата обращения 30.04.2024).
51. Overview of two-stage object detection algorithms. – URL: https://www.researchgate.net/publication/341871095_Overview_of_two-stage_object_detection_algorithms/ (дата обращения 30.04.2024).
52. Object Detection with RetinaNet. – URL: <https://keras.io/examples/vision/retinanet/> (дата обращения 30.04.2024).
53. Python ultralytics. – URL: <https://pypi.org/project/ultralytics/> (дата обращения 30.04.2024).
54. Object Detection: YOLOv1. – URL: <https://github.com/pjh5672/YOLOv1/> (дата обращения 30.04.2024).
55. YOLO9000: Better, Faster, Stronger. – URL: <https://arxiv.org/abs/1612.08242/> (дата обращения 30.04.2024).
56. Официальный Гитхаб YOLOv2. – URL: <https://github.com/Arup276/YOLOv2/> (дата обращения 30.04.2024).
57. Darknet53. – URL: <https://github.com/developer0hye/PyTorch-Darknet53/> (дата обращения 30.04.2024).
58. ArXiv is a free distribution service and an open-access archive. – URL: <https://arxiv.org/> (дата обращения 30.04.2024).
59. Официальный Гитхаб YoloV5. – URL: <https://github.com/ultralytics/yolov5> (дата обращения 30.04.2024).
60. Официальный Гитхаб YoloV6. – URL: <https://github.com/meituan/YOLOv6> (дата обращения 30.04.2024).
61. HIT UAV Infrared Thermal Dataset. – URL : <https://github.com/suojiaashun/HIT-UAV-Infrared-Thermal-Dataset> (дата обращения 30.04.2024).
62. Официальный сайт Docker. – URL: <https://www.docker.com/> (дата обращения 30.04.2024).

63. Docker Desktop WSL 2 backend on Windows. – URL: <https://docs.docker.com/desktop/wsl/> (дата обращения 30.04.2024).

64. Алферьева Т. И. Методические рекомендации для подготовки и оформления выпускных квалификационных работ: методические рекомендации для обучающихся всех форм обучения укрупненной группы направлений подготовки 09.00.00 Информатика и вычислительная техника / Т. И. Алферьева, В. Н. Васина, Д. Б. Шадрин. – Екатеринбург: УрФУ, 2019. – 78 с. – URL: <https://study.urfu.ru/Aid/ViewMeta/13865> (дата обращения 02.05.2024)

65. Репозиторий кода дипломного проекта на ресурсе GitHub. – URL: https://github.com/semyon-v-dot/aero_detect (дата обращения 30.04.2024).