**Министерство науки и высшего образования**

**Российской Федерации**

**ФЕДЕРАЛЬНОЕ Государственное Автономное**

**образовательное учреждениевысшего образования**

**«новосибирский национальный исследовательский государственныйуниверситет»**

**ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНОЙ РОБОТОТЕХНИКИ**

Кафедра Интеллектуальных систем теплофизики ИИР

Направление подготовки 15.03.06 Мехатроника и робототехника

Направленность (профиль) Мехатроника и робототехника

**ОТЧЕТ**

**о прохождении производственной практики, технологической (проектно-технологической) практики**

(указывается наименование практики)

**Обучающегося \_\_\_**Емельянова Алексея Алексеевича\_\_**\_\_\_\_ группы №** \_20931\_ 4 **курса**

(Ф.И.О. полностью)

**Тема задания**: \_\_\_ Разработка программного обеспечения для детекции людей с беспилотного летательного аппарата на основе технологий машинного зрения\_\_\_

**Место прохождения практики:** \_\_\_Федеральное государственное автономное образовательное учреждение высшего образования «Новосибирский национальный исследовательский государственный университет», 630090, Новосибирская область, г. Новосибирск, ул. Пирогова, д. 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное наименование организации и структурного подразделения, индекс, адрес)

**Сроки прохождения практики:** с 27.02.2025 г. по 17.03.2025 г.

**Руководитель практики от НГУ** Галактионова Ю.Ю., Специалист УМО \_\_\_\_\_\_\_\_

(Ф.И.О. полностью, должность) (подпись)

**Руководитель ВКР** Яковлев Дмитрий Александрович, Программист \_\_\_\_ \_\_\_\_\_\_\_\_

(Ф.И.О. полностью, должность) (подпись)

**Оценка по итогам защиты отчета:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(неудовлетворительно, удовлетворительно, хорошо, отлично)

**Отчет заслушан на заседании кафедры**   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ КафИСТИИР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование кафедры)

**протокол \_\_\_\_\_\_\_\_\_от** «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_г.

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ 3](#_Toc193090496)

[ВВЕДЕНИЕ 4](#_Toc193090497)

[1 АРХИТЕКТУРА КОДА ML и DL МОДЕЛЕЙ, ИСПОЛЬЗУЕМЫХ ДЛЯ РЕШЕНИЯ 5](#_Toc193090498)

[1.1 Виды существующих решений 5](#_Toc193090499)

[1.1.2 YOLOv8 5](#_Toc193090500)

[1.1.2 YOLO11 7](#_Toc193090501)

[1.1.3 DETR 8](#_Toc193090502)

[2.1 Стратегия обучения 10](#_Toc193090503)

[2.2 YOLO 11](#_Toc193090504)

[2.2 DETR 12](#_Toc193090505)

[2.3 Показатели эффективности 12](#_Toc193090506)

[2.3.1 Моментальные показатели 12](#_Toc193090507)

[2.3.2 Интегральные показатели 13](#_Toc193090508)

[2 РЕЗУЛЬТАТЫ 15](#_Toc193090509)

[2.1 Проверка датасетов на сбалансированность 15](#_Toc193090510)

[2.2 Результаты использования моделей 16](#_Toc193090511)

[2.2.1 YOLO 16](#_Toc193090512)

[2.2.2 DETR 18](#_Toc193090513)

[ЗАКЛЮЧЕНИЕ 20](#_Toc193090514)

[СПИСОК ЛИТЕРАТУРЫ 21](#_Toc193090515)

[ПРИЛОЖЕНИЕ А 22](#_Toc193090516)

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И ТЕРМИНОВ

БПЛА — это беспилотный летательный аппарат, воздушное судно без экипажа на его борту.

Детекция — обнаружение или нахождение какого-либо объекта.

Датасет — это структурированный набор данных, который содержит в себе конкретный набор свойств: признаки, связь с объектами или расположение в выборке данных. Используются данные для обучения нейронных сетей.

Гиперпараметры — параметры, которые управляют процессом обучения модели и настраиваются до начло процесса обучения.  
 Аугментация — набор методов и техник позволяющие искусственно расширить обучающеюся выборку модифицируя исходные данные в процессе обучения.

NLP (Natural Language Processing) — это направление в машинном обучении, посвящённое распознаванию, генерации и обработке устной и письменной человеческой речи.

Эмбеддинги — это способ представления данных в виде числовых векторов

# ВВЕДЕНИЕ

Проблема поиска пропавших людей в лесистой местности и труднодоступных районах остается актуальной на протяжении многих лет. Ежегодно фиксируются многочисленные случаи исчезновения туристов, грибников, детей и пожилых людей в дикой природе, что требует оперативного реагирования с целью предотвращения трагических последствий.

Современные технологии открывают новые возможности для решения данной проблемы. Одним из перспективных направлений является использование беспилотных летательных аппаратов (БПЛА) с системой автоматического обнаружения людей. Применение БПЛА позволяет эффективно обследовать обширные территории за короткий промежуток времени, а использование методов машинного зрения значительно повышает точность детекции, снижая вероятность пропуска объектов. Существует большое количество готовых алгоритмов для решения разных задач, но не каждая из них может подойти для детекции потерявшихся людей в труднодоступных местах. Именно поэтому важно понимать какие нейронные сети стоит применять и то, как модифицировать их.

Цель данной работы заключается в анализе существующих решений по детекции людей в труднодоступных местах и применение их для детекции людей.

# 1 АРХИТЕКТУРА КОДА ML и DL МОДЕЛЕЙ, ИСПОЛЬЗУЕМЫХ ДЛЯ РЕШЕНИЯ

## 1.1 Виды существующих решений

Современные методы машинного и глубокого обучения играют ключевую роль в задаче обнаружения людей с БПЛА. Использование нейронных модулей, в архитектуре, определяется специфичной задачей. Для решения задачи были применено несколько моделей. Существует несколько широко применяемых архитектур моделей для детекции объектов, каждая из которых обладает своими преимуществами и недостатками.

### 1.1.2 YOLOv8

YOLOv8 (You Only Look Once) — это современная модель для решения задач компьютерного зрения [1]. Она является частью семейства YOLO. YOLO — одна из наиболее популярных архитектур глубокого обучения для обнаружения объектов. Особенностью данной модели заключается в том, что она выполняет предсказание координат объектов и их классов за один проход через нейросеть. Благодаря этому модель обеспечивает высокую скорость работы. YOLOv8 является улучшенной версией YOLOv5. Архитектура YOLOv8 изображена на рисунке 1.

Архитектуру YOLOv8 можно условно разделить на три основных компонента:

* Backbone — это сверточная нейронная сеть (CNN), отвечающая за извлечение признаков из входного изображения. YOLOv8 использует специальную CSPDarknet53 backbone, которая использует кросс-ступенчатые частичные соединения для улучшения потока информации между слоями и повышения точности.
* Шея — шея, также известная как экстрактор признаков, объединяет карты признаков с разных этапов позвоночника для сбора информации в разных масштабах. Архитектура YOLOv8 использует новый модуль C2f вместо традиционной сети пирамид признаков (FPN). Этот модуль объединяет высокоуровневые семантические признаки с низкоуровневой пространственной информацией, что приводит к повышению точности обнаружения, особенно для небольших объектов.
* Голова — отвечает за прогнозирование. YOLOv8 использует несколько модулей обнаружения, которые прогнозируют ограничивающие рамки, оценки объектности и вероятности классов для каждой ячейки сетки на карте признаков. Затем эти прогнозы объединяются для получения окончательных обнаружений.

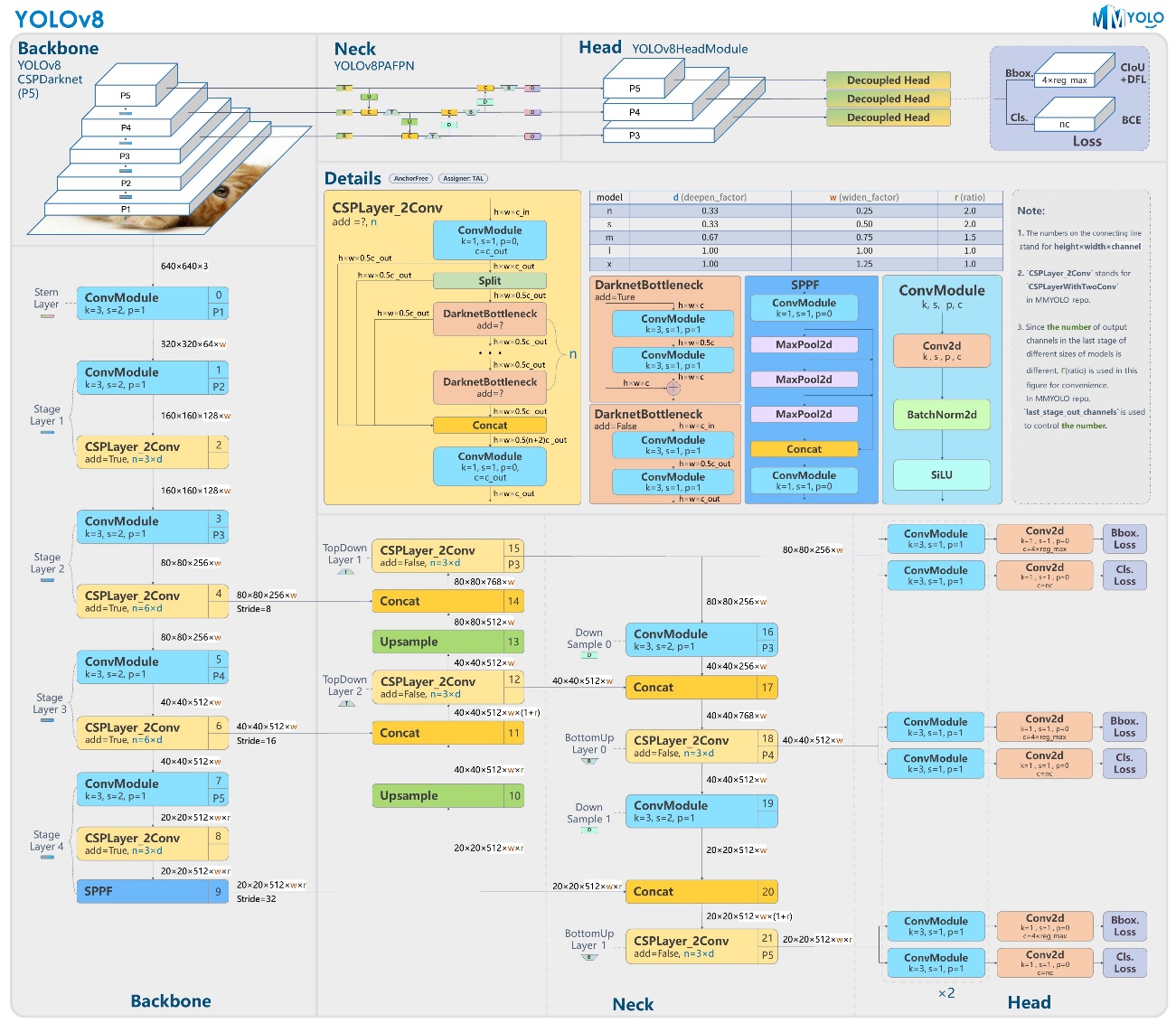


Рисунок 1 – Архитектура YOLOv8.

Преимущество YOLOv8

* Высокая точность: YOLOv8 достигает высочайшей точности в различных тестах обнаружения объектов.
* Скорость в реальном времени: модель демонстрирует впечатляющую скорость вывода, что делает ее пригодной для приложений реального времени, таких как автономные транспортные средства и робототехника.
* Эффективность: YOLOv8 имеет малый вес и требует меньше вычислительных ресурсов, чем другие модели, что делает его идеальным для развертывания на периферийных устройствах.
* Открытый исходный код и поддержка сообщества: YOLOv8 имеет открытый исходный код и поддерживается активным сообществом, что способствует постоянному развитию и совершенствованию архитектуры YOLOv8.

Недостатки YOLOv8

* Может давать неточные предсказания для мелких объектов.
* Склонность к пропуску объектов при сложных условиях освещения.

### 1.1.2 YOLO11

YOLO11 — это обновление архитектуры YOLOv8 с некоторыми новыми интеграциями и настройкой параметров [2]. Данная версия не является официальной и больше несет экспериментальный характер. Данная архитектура изображена на рисунке 2. Отличия архитектуры YOLO11 от YOLOv8:

* В шее блок C2f был заменен на C3k2. Это позволило быстрее обрабатывать данные при сохранении высокой производительности так как использует две свертки вместо одной большой свертки [2].
* Механизм внимания. После блока SPPF добавлен новый блок C2PSA. Он помогает модели сосредоточиться на ключевых областях изображения для лучшего обнаружения. Это отсутствует в YOLOv8, что делает YOLO11 потенциально более точным в обнаружении более мелких или закрытых объектов [3].

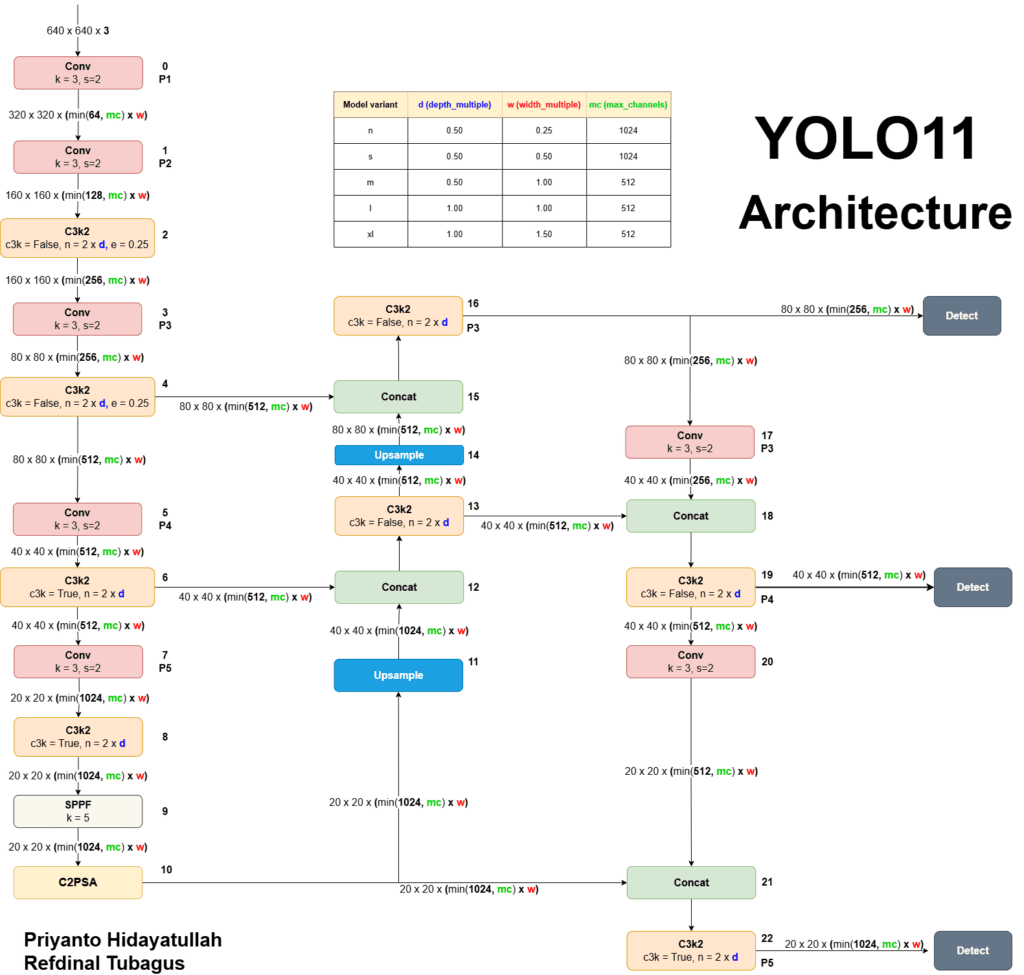
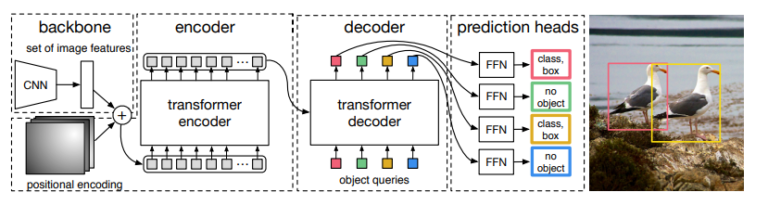


Рисунок 2 – Архитектура YOLO11.

### 1.1.3 DETR

DETR (DEtection TRansformer) — это архитектура отличающиеся от традиционных методов детекции объектов, основанных на сверточных нейронных сетях с механизмом якорей и постобработкой, DETR применяет трансформерную архитектуру. Главной особенностью данной архитектуры является его способность выполнять детекции объектов как задачу прямого сопоставления между входными признаками изображения и предсказанными ограничивающемся рамками, без генерации якорей и сложных постобработок [4]. Архитектура DETR изображена на рисунке 3.

Рисунок 3 – Архитектура DETR.

Общая структура DETR:

1. Backbone — На первом этапе изображение обрабатывается классическим сверточным экстрактором признаков, например, ResNet-50. Он преобразует входное изображение в пространственно-сверточное представление [5].
2. Transformer Encoder-Decoder — ключевая часть модели трансформер, применяемой в NLP. Encoder преобразует признаки изображения, добавляя позиционные эмбеддинги для сохранения пространственной информации. Decoder использует фиксированное количество запросов (object queries), каждый из которых обучается находить конкретный объект на изображении. Эти запросы взаимодействуют с закодированными признаками через механизм внимания (multi-head attention), что позволяет выявить потенциальные объекты [5].
3. Prediction Head — выход декодера направляется в два независимых линейных блока, предсказывающий координаты ограничивающей рамки и класс обнаруженного объекта [5].

Данная архитектура позволяет захватывать сложные взаимосвязи между объектами и их пространственным контентом, одновременно предсказывает класс и ограничивающие рамки для всех объектов на изображении, но данных подход имеет недостатки в скорости обучения по сравнению с YOLO, требует значительное количество ресурсов и плохо работает с мелкими объектами.

## 2.1 Стратегия обучения

Для обучения модели все исходные данные были предварительно обработаны и разделены на три выборки: обучающую (70 %), валидационную (20 %) и тестовую (10 %). Такой подход обеспечивает корректную оценку качества модели на различных этапах её обучения и последующей проверки.

Изображения в исходных датасетах имели высокое разрешение (3840 × 2160 пикселей), что оказывает существенное влияние как на процесс обучения нейронной сети, так и на конечные результаты. В связи с этим было принято решение разбить изображения на фрагменты размером 640 × 640 пикселей с перекрытием в 10 пикселей. Это позволило избежать потери объектов, расположенных в зонах разреза изображения, и сохранить соответствующие аннотации для каждой из частей. Такой подход не только расширяет обучающую выборку, но и способствует более точному обнаружению мелких объектов с высоты полета БПЛА. Так же, он позволяет компенсировать ограничения архитектур моделей DETR и YOLO при работе с мелкомасштабными объектами.

В качестве инструментария для обучения была использована библиотека Ultralytics, предоставляющая гибкую настройку гиперпараметров и простоту интеграции с различными архитектурами детекторов объектов. В процессе экспериментов были выбраны и настроены гиперпараметры, обеспечивающие наилучший баланс между точностью и производительностью модели.

Поскольку задача направлена исключительно на обнаружения одного класса — человек, обучение модели в режиме одного класса (single\_cls = True) позволяет повысить точность и устойчивость к ложным срабатываниям на фоновые объекты. Модель фокусируется исключительно только на признаках, характерных для человека, без внимания на другие классы.

Пороговое значение IoU (Intersection over Union) установлено на уровне 0.4, что позволяет достичь баланса между строгой оценкой точности детекции и необходимостью минимизировать пропуски объектов. В условиях поиска человека при аварийно-спасательных операциях критически важно не упустить объект, даже если точность границ ограничена.

Несмотря на общее количество изображений, доля изображений с людьми после предобработки составляет порядка 15 %. Для компенсации данного дисбаланса применялась техника аугментации Copy-Paste с вероятностью 0.5, что позволило искусственно увеличить количество обучающих примеров и повысить устойчивость модели к различным условиям сцены.

### 2.2 YOLO

В рамках исследования для обучения модели были выбраны современные архитектуры YOLOv8 и YOLO11. Они обладаю высокой скоростью и точностью детектировании объектов. Из-за ограничения ресурсов устройства, на котором производилось обучение, были использованы следующие версии моделей: yolov8n, yolov8s, yolo11n и yolo11s. По ходу работы особое внимание уделялось подбору гиперпараметров.

Размер batch определяет количество изображений, обрабатываемых за один шаг обучения. При работе с изображением в 640х640px и ограничениями вычислительными ресурсами, значение 18 является оптимальным с точки зрения загрузки видеопамяти и устойчивости градиентов. Таким образом, данное значение не приведет к снижению обобщающей способности и не приведет к нестабильному обучению.

Параметр отключения мозаичной аугментации позволяет повысить стабильность детекции за счет обучения на более “реальных” изображениях. По ходу работы было принято решение отключать мозаичную аугментацию за 20 эпох до завершения обучения. Таким образом модель не только получит преимущество от разнообразия данных на начальных этапах, но и постепенно переходит к обучению на более приближенных к реальности изображениях, что способствует к лучшей адаптации к реальным условиям эксплуатации.

### 2.2 DETR

Модель DETR продемонстрировала высокую чувствительность к качеству обучающего набора и требовательность к вычислительным ресурсам. В связи с этим размер batch size был установлен на уровне 16 — оптимальное значение, обеспечивающее баланс между производительностью и стабильностью обучения при заданном объеме видеопамяти.

## 2.3 Показатели эффективности

Для оценки качества обучения модели для детектирования человека использовались как моментальные, так и интегральные показатели. Для оценки стабильности и надежности полученных показателей дополнительно был расчет средней ошибки среднего (SEM) — характеризует уверенность в оценке среднего значения метрики (2). Стандартное отклонение (SD) — показывает меру разброса значений метрик при повторных запусках модели (1).

(1)

Где a — среднее значение показателя, — отдельные наблюдения, n — количество наблюдений.

(2)

Где n — количество наблюдений. Низкое значение SEM указывает на высокую точность оценки метрики.

### 2.3.1 Моментальные показатели

Моментальные показатели позволяют оценить, полноту и устойчивость при выполнении задачи детекции одного класса. При решении задачи детекции человека особое внимание было уделено таким метрикам как:

* Precision (Точность положительных предсказаний) — показывает насколько модель уверенно детектирует именно человека и не путает его с фоновым объектом.

(3)

где, TP (True Positives) - количество верно предсказанных положительных примеров, которые были правильно классифицированы как положительные; FP (False Positives) - количество неверно предсказанных положительных примеров, которые были неправильно классифицированы как положительные.

* Recall (Чувствительность / полнота) — характеризует способность модели находить всех людей в кадре. Данная метрика играет очень важна, так как для решение задачи нужно находить максимальное количество людей в условиях поисково-спасательных операциях.

(4)

где: TP (True Positives) - количество истинно положительных примеров, которые были правильно классифицированы как положительные; FN (False Negatives) - количество ложно.

* Mean Average Precision (mAP): — показывает насколько предсказанные ограничивающие рамки совпадают с истинными.

(5)

где 𝐴— площадь под PR-кривой (Precision-Recall) для каждого объекта, N — количество объектов. Для оценки применялась mAP@0.5 и mAP@0.5:0.95, отражающие качество детекции при разных порогах IoU.

### 2.3.2 Интегральные показатели

Интегральные показатели отражают общую способность модели различать положительные и отрицательные классы в рамках всей области предсказаний. Особое внимание уделялось AUC (Area Under Curve) — площадь под ROC-кривой. ROC-кривая строится на основе соотношения **True Positive Rate (TPR)** и **False Positive Rate (FPR)** при изменении порога классификации. График ROC-кривой отображает зависимость между долей, верно, классифицированных положительных примеров TPR и долей ложно классифицированных отрицательных примеров FPR при варьировании порога классификации.

(6)

AUC отражает интегральную способность модели отличать положительный класс от отрицательного. Если AUC близится к значению 1, то это свидетельствует о высокой эффективности модели. Значение 0,5 несет случайный характер предсказаний.

(7)

# 2 РЕЗУЛЬТАТЫ

## 2.1 Проверка датасетов на сбалансированность

Перед началом обучения моделей была проведена оценка сбалансированности имеющегося датасета. Основной задачей являлась проверка равномерного распределения объектов по классам, а также выявление возможного дисбаланса, который бы мог повлиять на качество обучения.

В рамках рассматриваемой задачи использовался только один класс – человек. Оценка сбалансированности проводилась с точки зрения соотношения изображений, содержащих человека и изображений без него. Датасет “Lacmus” содержал изображение человека на каждом изображении, до предобработки данных. Так как в рамках исследования было принято разрезать картинки на мелкие части, чтобы избежать проблем нахождения мелких объектов, набор данных для обучения увеличился примерно в 21 раз, но стало больше изображений, где целевой объект отсутствует. Анализ показал, что в данном наборе данных примерно 49% изображений содержат человека.

В датасете “Аэрозрение” изначально присутствовали изображения, где нет человека, и после предобработки данных, было выявлено, что примерно 80% изображений в котором целевого объекта не было. Это говорит о некотором дисбалансе в пользу отрицательных примеров, что характерно для задач обнаружения редких объектов поисково-спасательных операциях с БПЛА. Для компромисса данного дисбаланса были применены следующие шаги:

* Решение проблемы через аугментации. Применение аугментации (copy-past) поможет расширить количество изображений с целевым объектом и использование режима обучения в одном классе, что позволит модели сфокусироваться только на признаках человека.
* Удаление из предобработанных данных 50% изображений, в котором человек не присутствует.

Таким образом, была обеспечена более равномерное представление целевого объекта, что способствует обобщающей способностью модели.

## 2.2 Результаты использования моделей

В результате обучения моделей были получении и проанализированы на эффективность архитектуры результаты моделей YOLOv8s, YOLOv8n, YOLO11s, YOLO11n и RTDETR-l.pt.

### 2.2.1 YOLO

На начальном этапе обучения модели проходили обучение на исходных полноразмерных изображениях (3840×2160 пикселей) без применения разрезания на фрагменты. Так же были применены аугментации, которые бы имитировали реалистичный условия палета дона:

* Повороты, изменение яркости и контрастности — для приближения условий к различному естественному освещению.
* Размытие (blur) — для имитации движения дрона во время съёмки.
* Искажение перспективы — учитывая, что камера БПЛА не всегда располагается строго перпендикулярно земле, а также возможны вибрации.
* MixUp — для моделирования ситуаций, когда человек частично перекрыт элементами окружающей среды, такими как деревья или кустарники. Результаты метрик на тестовом наборе данных при обучении на архитектурах семейства YOLO отражены в таблице 1.

Таблица 1 - Результаты сравнения моделей архитектур семейства YOLO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модель | Precision (±SD) | Recall (±SD) | F1-score  (±SD) | mAP@0.5  (±SD) | mAP@0.5:0.95  (±SD) |
| YOLOv8n | 0.95 ± 0.01 | 0.64 ± 0.02 | 0.76 ± 0.02 | 0.95 ± 0.02 | 0.53 ± 0.02 |
| YOLOv8s | 0.93 ± 0.02 | 0.74 ± 0.01 | 0.82 ± 0.02 | 0.87 ± 0.02 | 0.64 ± 0.03 |
| YOLO11n | 0.94 ± 0.02 | 0.62 ± 0.01 | 0.74 ± 0.02 | 0.94 ± 0.02 | 0.49 ± 0.02 |
| YOLO11s | 0.89 ± 0.02 | 0.72 ± 0.02 | 0.79 ± 0.02 | 0.85 ± 0.02 | 0.59 ± 0.04 |

Однако в процессе анализа выявилось, что обнаружение человека на полноразмерных изображениях затруднено, особенно в случаях, когда объект имеет небольшой масштаб на фоне общей сцены. Также значительная часть изображения зачастую не содержала целевых объектов, что снижало эффективность обучения и замедляло сходимость моделей.

Для решения данной проблемы была проведена предобработка данных — разрезание изображений с 3840×2160 на фрагменты 640×640 пикселей с перекрытием 10 пикселей. Это позволило:

* повысить плотность положительных примеров;
* улучшить обнаружение мелких объектов;
* уменьшить вычислительную нагрузку;
* повысить устойчивость и точность обучения моделей.

Такой подход также учитывался в этапе инференса. Для этого использовалась библиотека patched\_yolo\_infer, позволяющая подавать на вход модели разрезанные изображения, как это было реализовано при обучении.

После предобработки данных было зафиксировано значительное увеличение количества фрагментов, не содержащих целевого объекта (человека) — до 70 % от общего числа изображений. Для устранения возникшего дисбаланса классов дополнительно применялась аугментация Copy-Paste, при которой человек с одного изображения копировался на другое с вероятностью 50 %. Это позволило увеличить плотность положительных примеров и улучшить обучаемость модели.

Предобработка данных и применение аугментации оказали значительное положительное влияние на итоговые метрики моделей. Особенно заметное улучшение наблюдается в метрике Recall, что указывает на снижение числа пропущенных объектов. Разделение изображения на части позволило добиться более точного и надёжного распознавания человека, особенно в условиях мелкомасштабных объектов, характерных для съёмки с высоты БПЛА. Результаты моделей YOLO после предобработки представлены в Таблице 2.

Таблица 2 - Результаты работы моделей семейства YOLO после предобработки данных.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модель | Precision (±SD) | Recall (±SD) | F1-score  (±SD) | mAP@0.5  (±SD) | mAP@0.5:0.95  (±SD) |
| YOLOv8n | 0.93 ± 0.01 | 0.74 ± 0.02 | 0.84 ± 0.02 | 0.82 ± 0.02 | 0.63 ± 0.02 |
| YOLOv8s | 0.95 ± 0.02 | 0.84 ± 0.02 | 0.89 ± 0.02 | 0.92 ± 0.02 | 0.73 ± 0.02 |
| YOLO11n | 0.90 ± 0.01 | 0.74 ± 0.02 | 0.82 ± 0.02 | 0.80 ± 0.02 | 0.65 ± 0.02 |
| YOLO11s | 0.89 ± 0.02 | 0.86 ± 0.02 | 0.86 ± 0.02 | 0.90 ± 0.02 | 0.71 ± 0.02 |

### 2.2.2 DETR

Для сравнения эффективности различных архитектур была также протестирована модель DETR. Однако в процессе обучения данной модели возникли существенные сложности, связанные с её неспособностью адаптироваться к специфике представленного датасета и условиям задачи.

На этапе обучения DETR не продемонстрировала положительной динамики улучшения метрик. Наилучшая модель была зафиксирована уже на первой эпохе, при этом последующие эпохи не приводили к значимым улучшениям показателей точности.

Поскольку в процессе обучения был установлен механизм досрочного останова (early stopping) — если на протяжении 30 эпох не наблюдается улучшения метрик на валидационном наборе, обучение прекращается автоматически, — модель завершила обучение досрочно, после 30 эпох без прогресса.

Этот результат может свидетельствовать о том, что:

* архитектура DETR недостаточно устойчива к специфике задачи поиска мелких объектов на фрагментированных изображениях;
* требуется значительное увеличение объёма данных и более продолжительное обучение;
* либо модель требует другого подхода к подготовке данных, например, аннотаций в другом формате или тонкой настройки гиперпараметров.

Таким образом, в условиях текущего датасета и стратегии разрезания изображений на фрагменты, модель DETR не показала удовлетворительных результатов, что делает её малоэффективной для применения в поставленной задаче по сравнению с архитектурами семейства YOLO.

# ЗАКЛЮЧЕНИЕ

В ходе учебной практики была проведена работа по анализу и сравнению современных моделей компьютерного зрения, применяемых для задачи обнаружения человека с беспилотного летательного аппарата в затрудненных местах. Основное внимание было уделено изучению архитектур семейств YOLOv8, YOLOv11 и DETR. Работа включала в себя подготовку и предобработку данных, обучение моделей, анализ метрик качества и оценку влияния различных стратегий обучения на результаты.

На первом этапе был проведён обзор существующих архитектур, выявлены их сильные и слабые стороны, после чего выбран ряд моделей, обладающих высокой скоростью и точностью при выполнении задач детекции. Так же был анализ недостатков моделей с поиском мелких объектов и их решение путем предобработки данных. Это позволило улучшить обучение за счёт повышения плотности целевых объектов в кадре и уменьшения количества пустых участков изображения. Также были применены различные виды аугментаций, имитирующие реальные условия съёмки: повороты, изменение яркости и контрастности, размытие, искажение перспективы и метод MixUp. После разрезания и анализа полученных данных дополнительно использовалась аугментация Copy-Paste для увеличения количества положительных примеров и устранения дисбаланса.

Был произведен сравнительный анализ моделей на эффективность, где YOLO показал высокую стабильность и хорошие результаты, особенно после предобработки данных. Модели стали находить больше объектов что значительно важно для задачи поисково-спасательных отрядов. В то же время модель DETR не продемонстрировала приемлемой сходимости: наилучшие показатели были достигнуты уже на первой эпохе, и дальнейшее обучение не приводило к улучшению результатов.

# СПИСОК ЛИТЕРАТУРЫ

1 YOLOv8: Comprehensive Guide to State Of The Art Object Detection / [Электронный ресурс] // learnopencv.com: [сайт]. — URL: https://learnopencv.com/ultralytics-yolov8/ (дата обращения: 05.03.2025).

2 YOLO11: Redefining Real-Time Object Detection / [Электронный ресурс] // learnopencv.com: [сайт]. — URL: https://learnopencv.com/yolo11/ (дата обращения: 05.03.2025).

3 YOLO11 против YOLOv8: подробное сравнение / [Электронный ресурс] // docs.ultralytics.com: [сайт]. — https://docs.ultralytics.com/ru/compare/yolo11-vs-yolov8/ (дата обращения: 09.03.2025).

4 DETR: Бесконечная история / [Электронный ресурс] // habr.com: [сайт]. — https://habr.com/ru/companies/sberdevices/articles/847324/ (дата обращения: 09.03.2025).

5 Архитектура DETR / [Электронный ресурс] // blog.deepschool.ru: [сайт]. — https://blog.deepschool.ru/cv/detection/detr/ (дата обращения: 09.03.2025).

# ПРИЛОЖЕНИЕ А

Инференс для запуска моделей с автоматическим разрезанием изображения на части с перекрытием с использованием библиотеки patched\_yolo\_infer.

from ultralytics import YOLO  
from PIL import Image  
from patched\_yolo\_infer import (  
 MakeCropsDetectThem,  
 CombineDetections,  
 visualize\_results\_usual\_yolo\_inference,  
 visualize\_results,  
)  
import cv2  
import matplotlib.pyplot as plt  
from Module.showInferensModel import load\_labels, plot\_image\_with\_boxes  
PATH\_BEST = '../Weightts/yolov8s\_100\_aug.pt'  
PATH\_TEST\_IMAGES = './DataSet/MyData/Test/images'  
PATH\_TEST\_LABELS = './DataSet/MyData/Test/labels'  
SAVE\_RESULT = './Result/'  
model = YOLO(PATH\_BEST)

NAME = '100'  
results = model(f'{PATH\_TEST\_IMAGES}/{NAME}.jpg')

visualize\_results(  
 img=result.image,  
 confidences=result.filtered\_confidences,  
 boxes=result.filtered\_boxes,  
 classes\_ids=result.filtered\_classes\_id,  
 classes\_names=result.filtered\_classes\_names,  
 thickness=8,  
 show\_boxes=True,  
 delta\_colors=3,  
 show\_class=False,  
 axis\_off=False)