



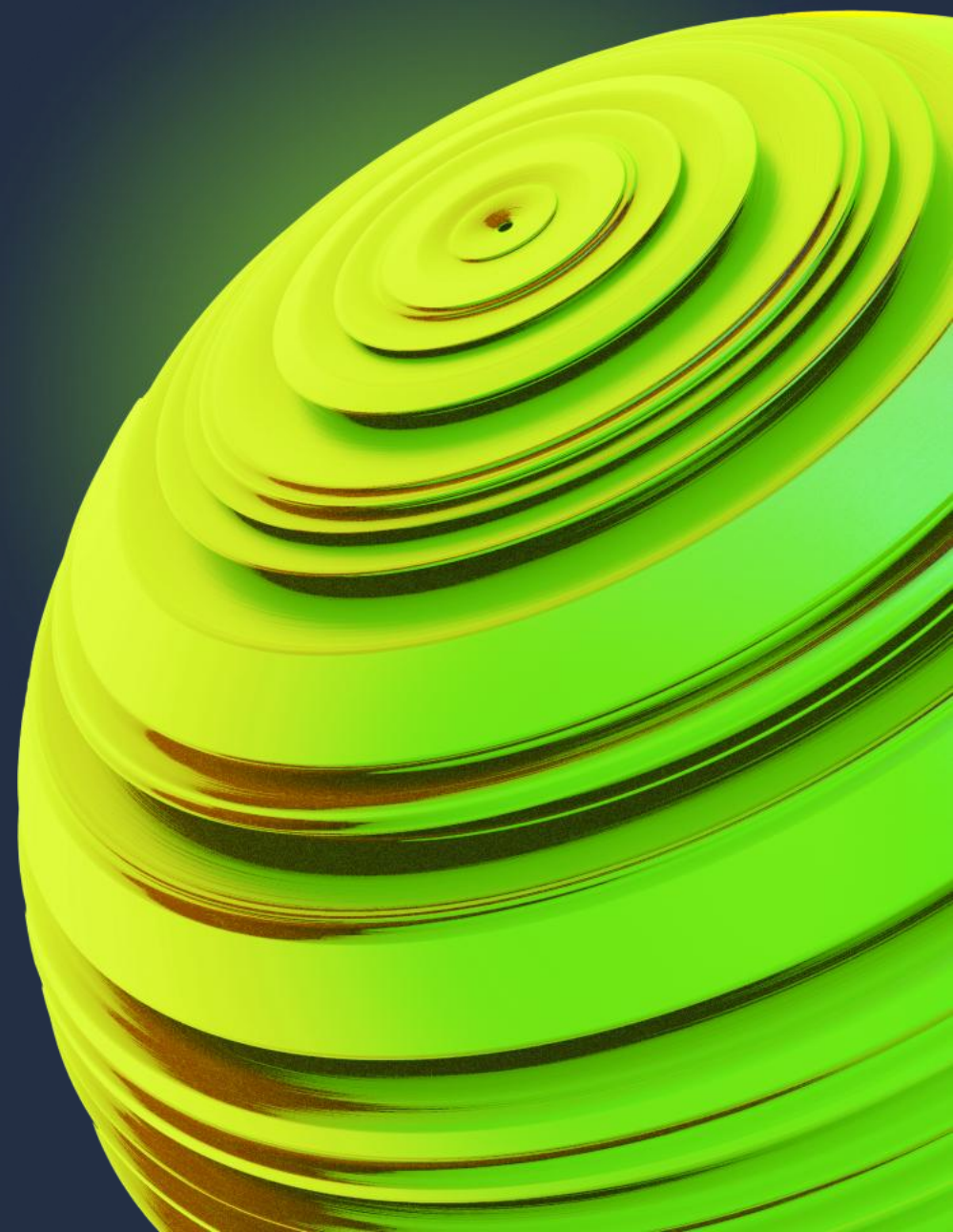
ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС



УНИВЕРСИТЕТ
ИННОПОЛИС

Итоговый проект по курсу ML-инженер

Детекция и классификация дорожных знаков





Цель и задачи проекта

Цель

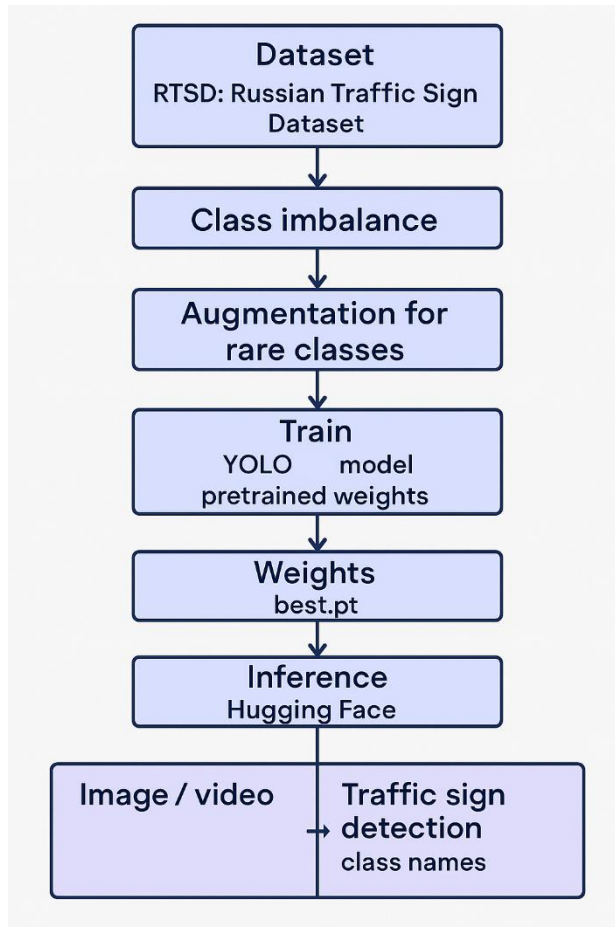
Разработать приложение, которое обнаруживает и классифицирует дорожные знаки по [ПДД РФ](#).

Задачи

- Выбрать подходящий датасет и модель.
- Провести первичный анализ данных.
- Подготовить датасет для обучения.
- Подобрать и обучить модель.
- Сделать приложение при помощи Gradio или Streamlit для обнаружения и классификации дорожных знаков.



Диаграмма пайплайна обучения и инференса модели



Исходный датасет

Датасет взят с платформы Kaggle - [RTSD \(Russian Traffic Sign Dataset\)](#)

- Содержит видеок cadры дорожной обстановки (разрешение 1280×720 – 1920×1080), снятые с бортового регистратора в разных сезонах, в разное время суток и при разной погоде
- 179138 изображений в датасете.
- 59188 аннотаций (размеченных знаков).
- 155 уникальных классов (наименований) знаков.
- Разметка типа 2_1 означает Знак 2.1 по ПДД РФ – «Главная дорога»

Обработанная разметка



Исходная разметка

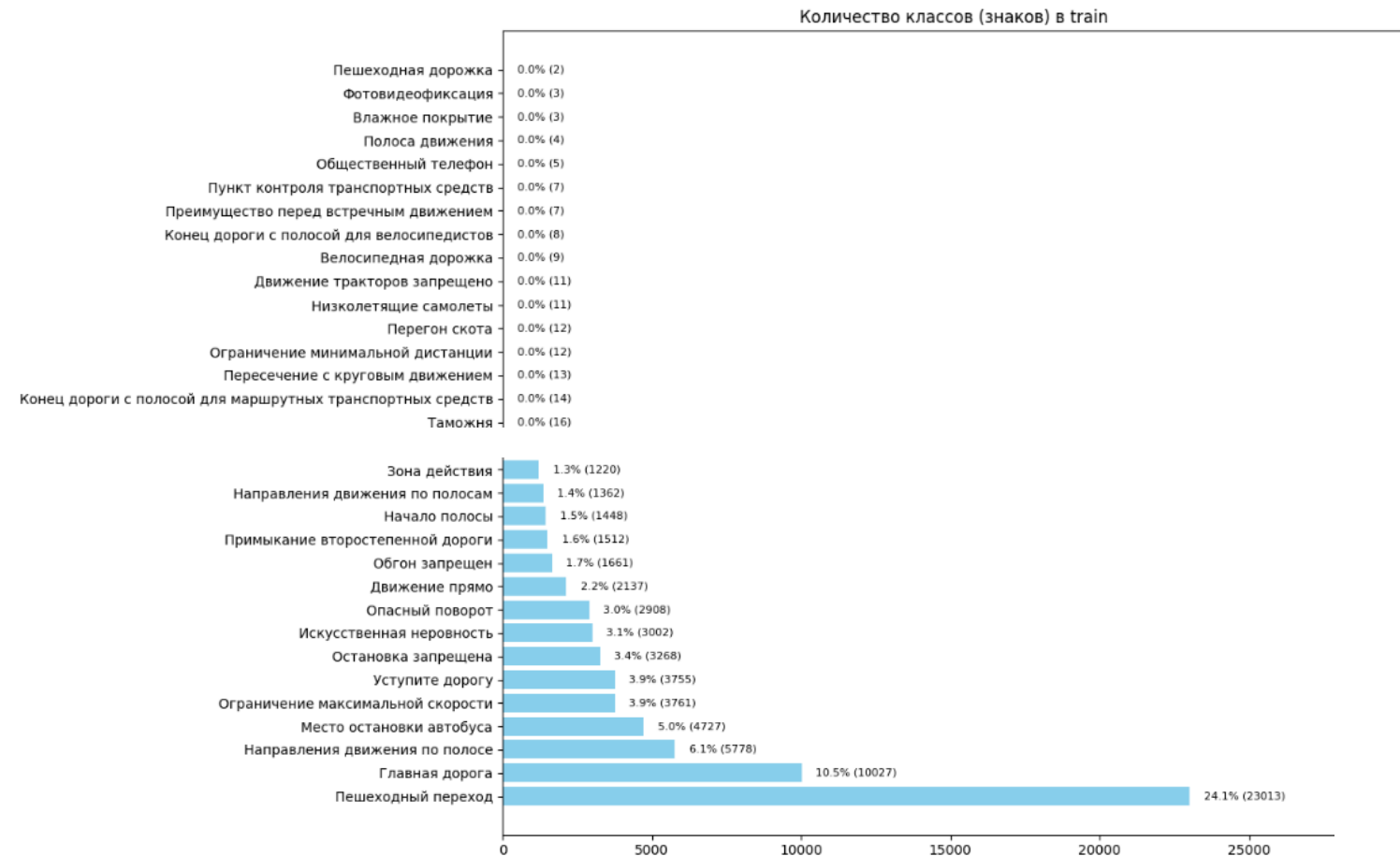




Исходный датасет. Дисбаланс классов.

Датасет имеет большой дисбаланс классов.

- Самый распространённый знак – «Пешеходный переход» (23013 боксов)
- 121 из 155 знаков находятся за чертой менее 500 экземпляров.
- Для увеличения экземпляров знаков было принято решение использовать аугментации для редких классов (менее 500).



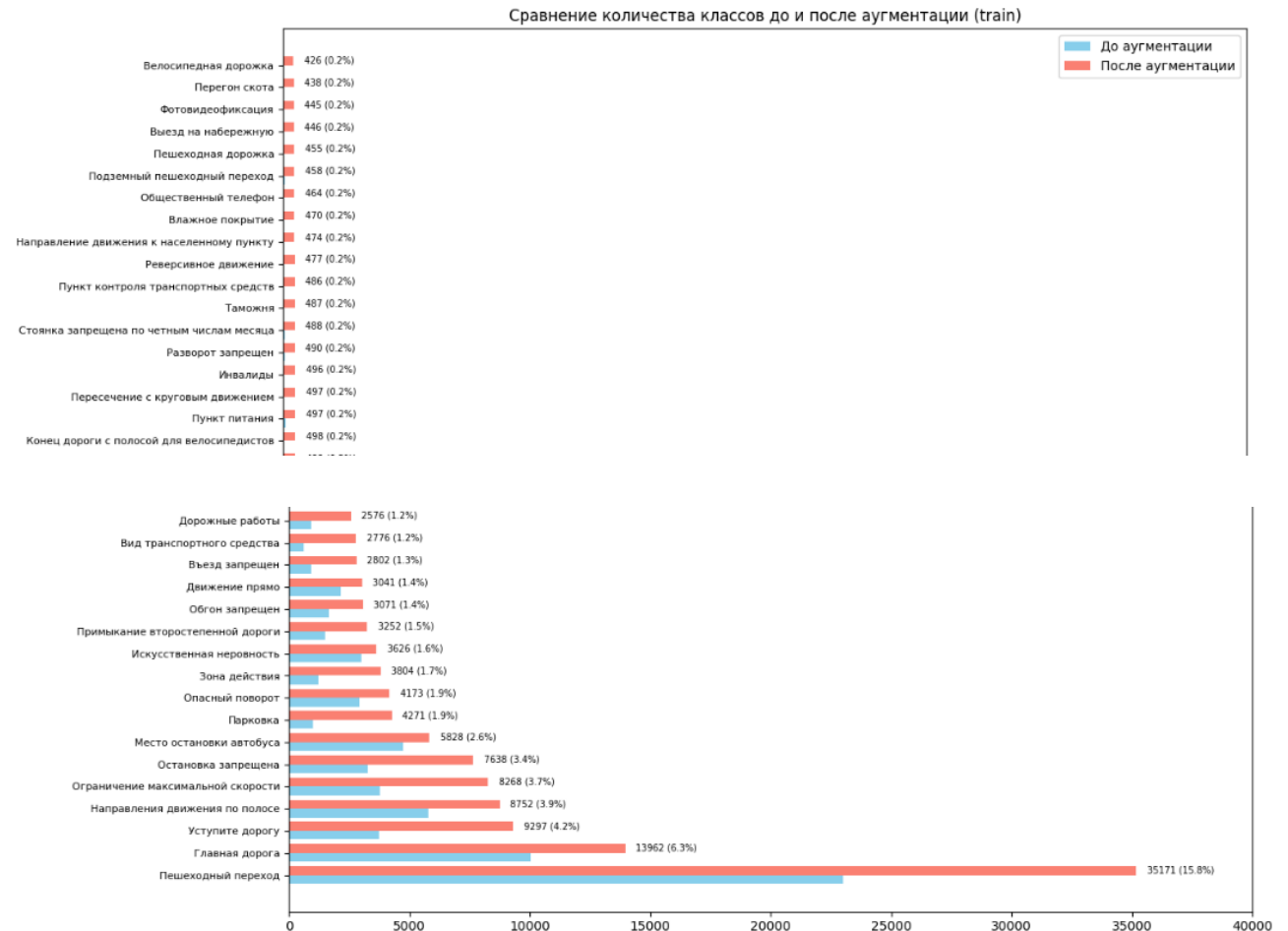


Борьба с дисбалансом классов.

Были применены Albumentations аугментации.

- Rotate,
- RandomBrightnessContrast,
- CLAHE,
- HorizontalFlip,
- Perspective
- Affine.

Видим, что подросли не только редкие классы, но и классы, у которых изначально было больше 500 боксов. Это связано с тем, что на одном изображении может быть несколько знаков, поэтому применяя аугментацию, мы меняем положение бокса как для редкого класса, так и для всех остальных классов, которые есть на изображении.



Примеры изображений после аугментации

aug_74_17336.jpg



aug_140_44887.jpg



aug_116_34636.jpg



aug_91_24029.jpg





Модель YOLO

В качестве обучаемых моделей были выбраны [YOLO11s](#) и [YOLO12s](#).

Для обучения моделей YOLO необходимо привести данные в такой формат:

YOLO_FORMAT/

```
|— images/  
|   |— train / ← данные с аугментацией  
|   |— val /   ← данные без аугментации  
|— labels/  
    |— train /  
    |— val /
```



input

YOLOv10

YOLOv11

YOLOv12

Были выбраны модели YOLO, так как нам необходимо быстро и на лету обнаруживать и классифицировать дорожные знаки.

Модель YOLO11 использует улучшенную backbone и neck-архитектуру, что повышает возможности извлечения признаков для более точного обнаружения объектов и выполнения сложных задач.

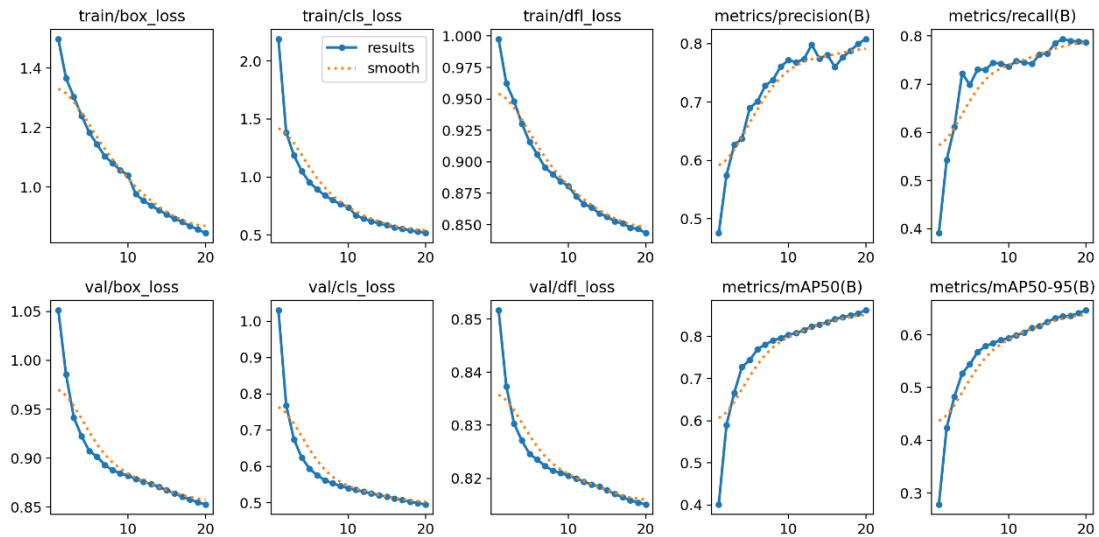
Модель YOLO12 представляет архитектуру, ориентированную на внимание, которая отходит от традиционных подходов на основе CNN, используемых в предыдущих моделях YOLO, но при этом сохраняет скорость инференса в реальном времени, необходимую для многих приложений. Эта модель достигает современной точности обнаружения объектов благодаря новым методологическим инновациям в механизмах внимания и общей архитектуре сети, сохраняя при этом производительность в реальном времени.



Обучение Моделей YOLO

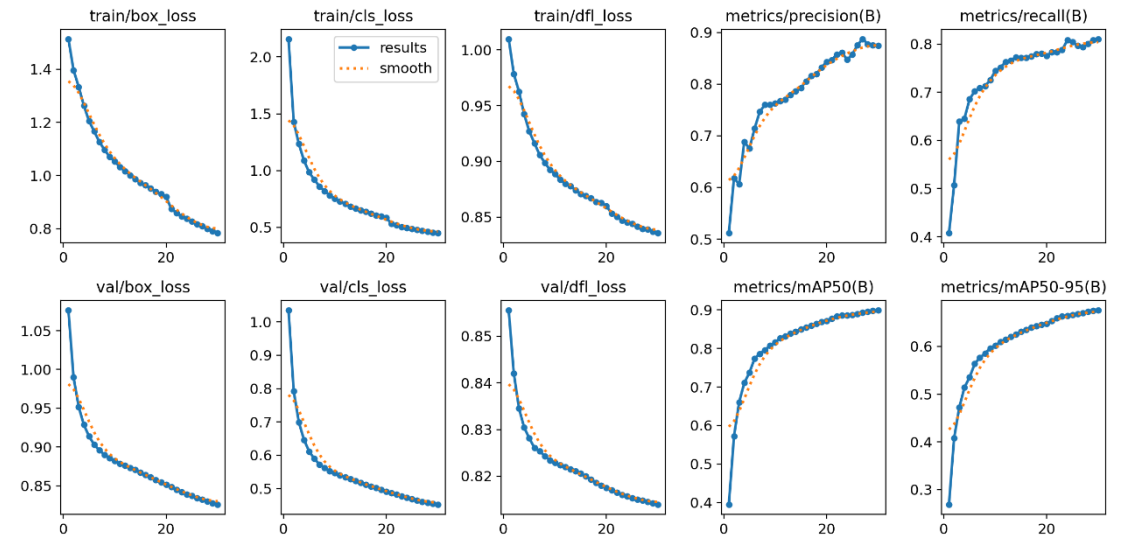
Модель [YOLO11s](#) обучалась на 20 эпохах (15 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB



Модель [YOLO12s](#) обучалась на 30 эпохах (28,5 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB

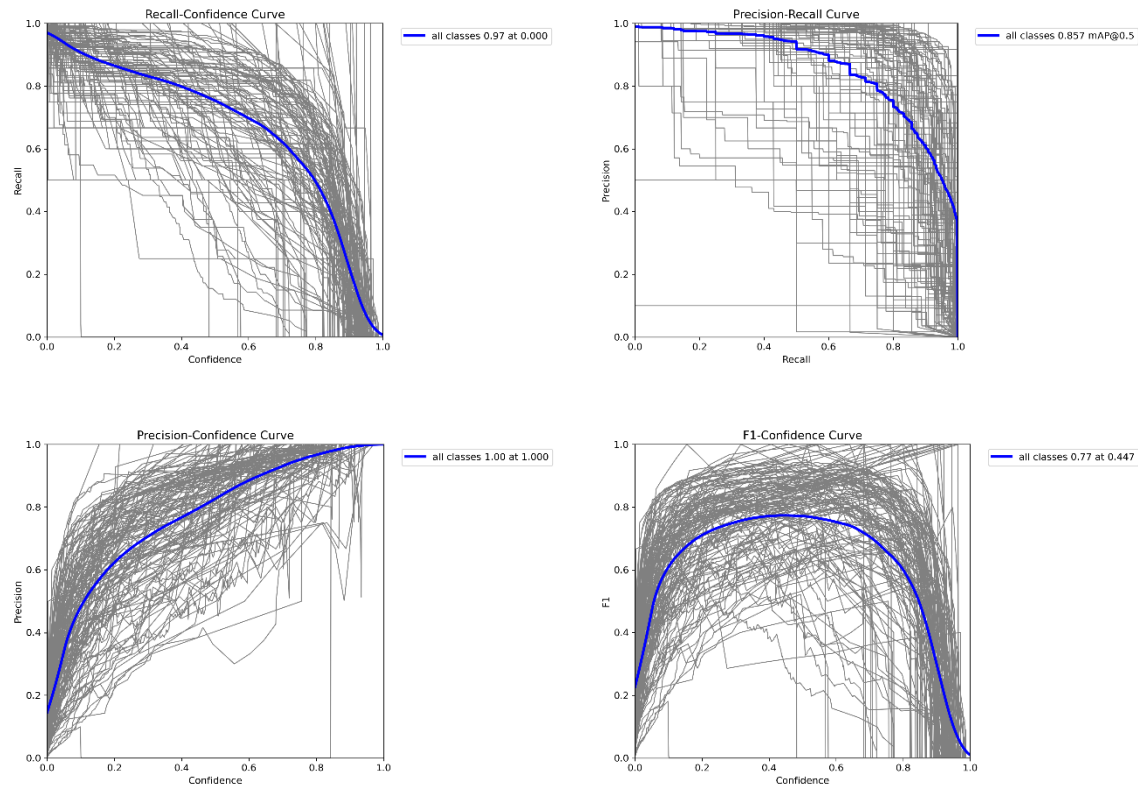




Обучение Моделей YOLO

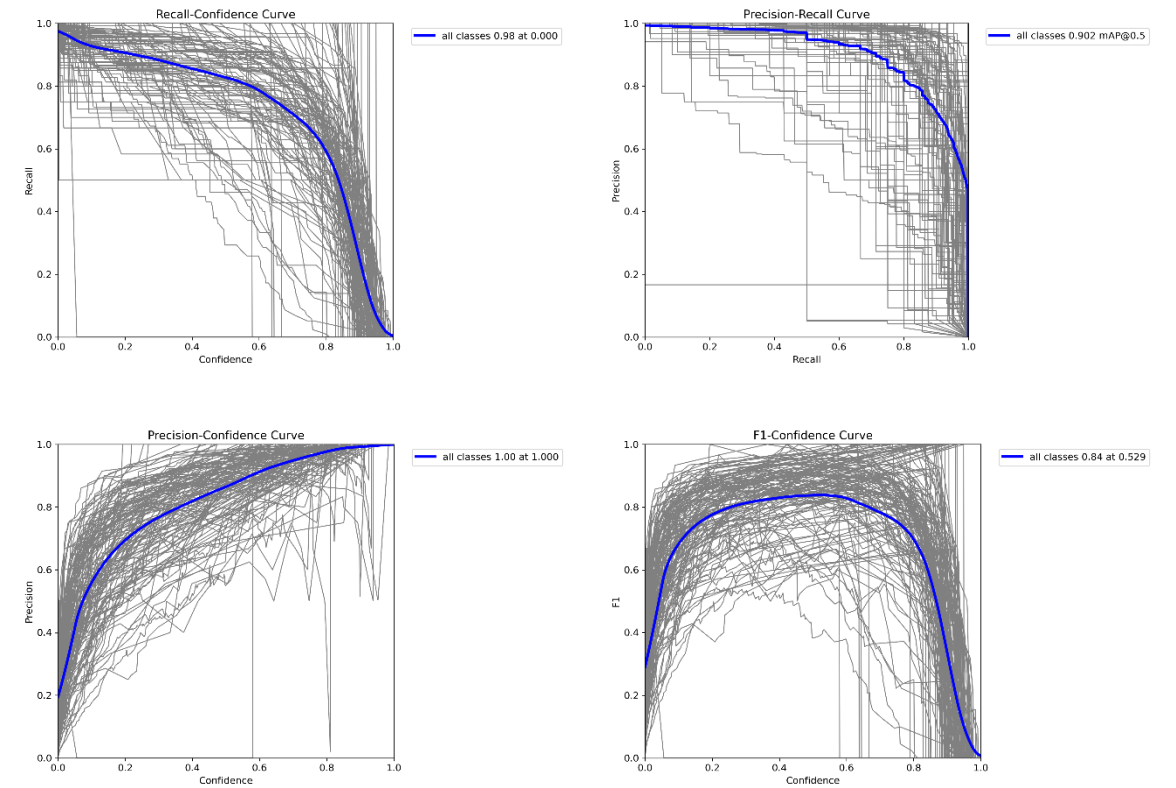
Модель [YOLO11s](#) обучалась на 20 эпохах (15 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB



Модель [YOLO12s](#) обучалась на 30 эпохах (28,5 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB





Обучение Моделей YOLO

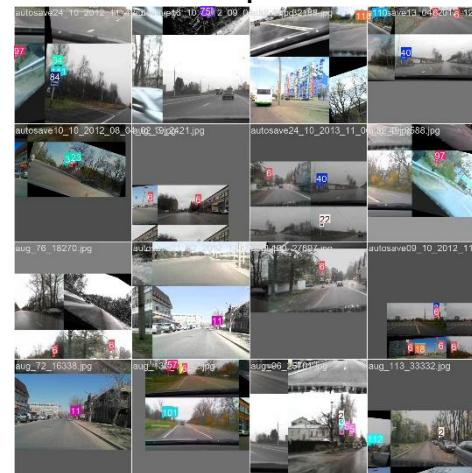
Модель YOLO11s обучалась на 20 эпохах (15 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB



--- Val batch labels

Train batch



--- Val batch pred



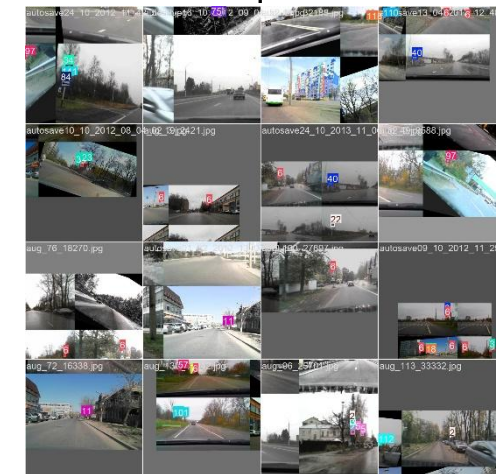
Модель YOLO12s обучалась на 30 эпохах (28,5 часов)

На видеокарте T4 (Colab) + GeForce RTX 3060 12GB

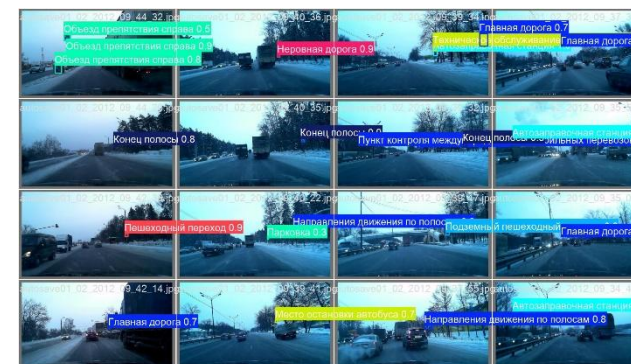


--- Val batch labels

Train batch



--- Val batch pred





Демо-приложение на Gradio

Для инференса была выбрана модель YOLO11s('best.pt').

Было написано приложение [Traffic Sign Detection With YOLO](#) с помощью библиотеки [Gradio](#) и загружено на сайт [Hugging Face](#)

Traffic Sign Detection With YOLO

[Readme](#) [Image](#) [Video](#)

Traffic Sign Detection With YOLO

Это интерактивное приложение на основе Gradio, которое позволяет выполнять детекцию дорожных знаков на изображениях и видео с использованием модели YOLO.

Стек технологий

- YOLO (Ultralytics) — модель детекции и классификации объектов
- PyTorch — фреймворк глубокого обучения
- OpenCV — обработка изображений и видео
- Gradio — веб-интерфейс для ML-приложений
- Pillow (PIL) — работа с изображениями
- FFmpeg — обработка аудио и видео

Структура приложения

- Загрузка изображений и видео
- Запуск детекции с настраиваемыми параметрами (порог, толщина, шрифт)
- Сохранение аннотированных файлов
- Предпросмотр результата
- Генерация текстовой легенды сбоку

Как пользоваться

- Откройте вкладку **Image** или **Video**
- Загрузите файл (изображение или видео)
- Настройте параметры:
 - Confidence Threshold - Порог уверенности
 - Box Thickness - Толщина рамки
 - Font Size - Размер шрифта
- Нажмите **Run** для предпросмотра
- Нажмите **Save** для сохранения результата
- Нажмите **Clear** для очистки полей

Примеры




- Example Images** — пример изображения дорожного знака
- Example Videos** — пример видео с реальными условиями

Traffic Sign Detection With YOLO

[Readme](#) [Image](#) [Video](#)


Upload Image

Перетащите изображение сюда
- или -
Нажмите для загрузки



Preview

Example Images



Confidence Threshold

0

1

0.5

Box Thickness

1

10

2

Font Size

8

72

18

Save Image File

Run Image

Save Image




Clear

Traffic Sign Detection With YOLO

[Readme](#) [Image](#) [Video](#)


Upload Video

Перетащите видео сюда
- или -
Нажмите для загрузки



Preview

Example Videos



Confidence Threshold

0

1

0.5

Box Thickness

1

10

2

Font Size

8

72

18

Save Video File

Run Video

Save Video

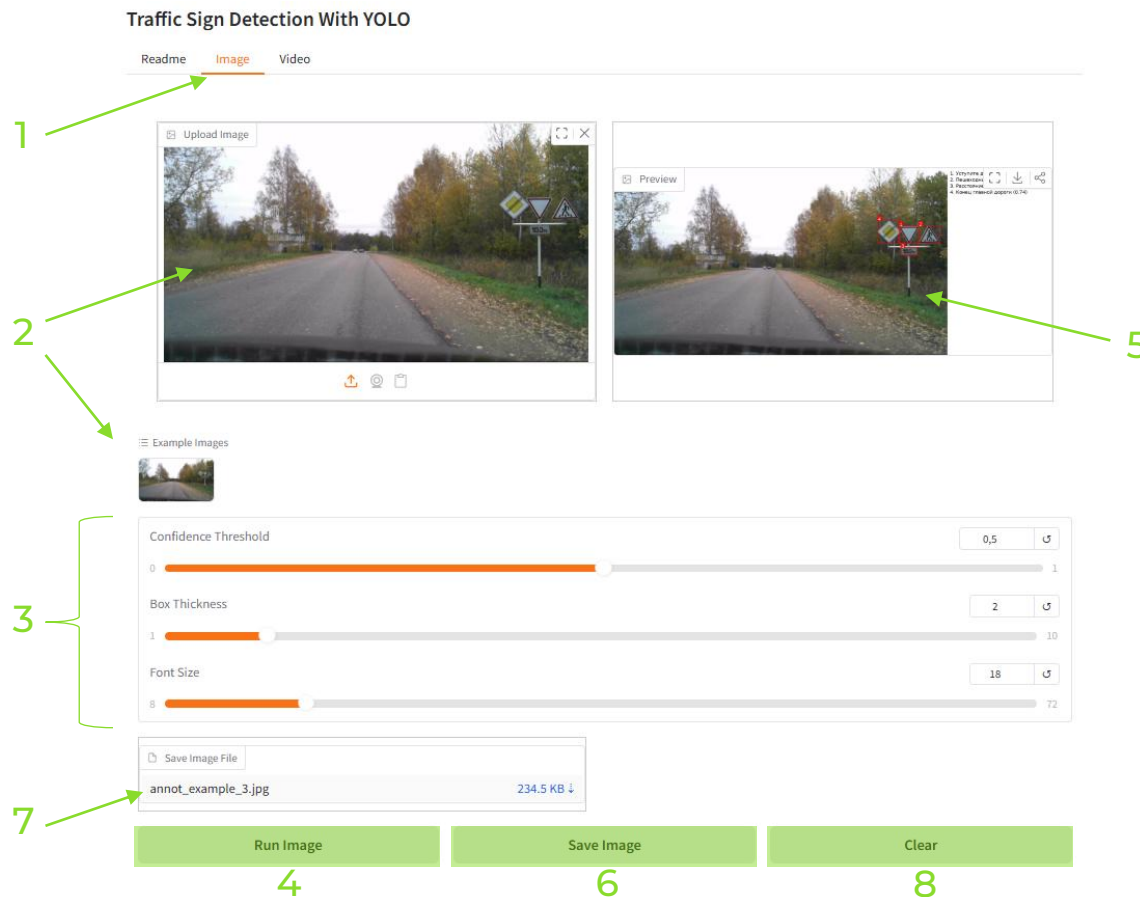
Clear



Демо-приложение на Gradio

Для инференса была выбрана модель YOLO11s('best.pt').

Было написано приложение [Traffic Sign Detection With YOLO](#) с помощью библиотеки [Gradio](#) и загружено на сайт [Hugging Face](#)



Как пользоваться приложением

1. Откройте вкладку **Image** или **Video**.
2. Загрузите файл (изображение или видео), используя загрузчик или веб-камеру. Для примера можно использовать *Example Images* или *Example Videos*.
3. Настройте параметры:
 - *Confidence Threshold* - Порог уверенности
 - *Box Thickness* - Толщина рамки
 - *Font Size* - Размер шрифта
4. Нажмите *Run Image* или *Run Video* для предпросмотра.
5. В окне предпросмотра (*Preview*) появится результат детекции и классификации.
6. Нажмите *Save Image* или *Save Video* для сохранения результата.
7. В окне *Save Image File* или *Save Video File* появится ссылка для скачивания.
8. Нажмите *Clear* для очистки полей.



Общие выводы

Для решения поставленных задач были выполнены следующие шаги:

- Был выбран датасет [RTSD \(Russian Traffic Sign Dataset\)](#).
- Для борьбы с дисбалансом классов были применены аугментации редких классов.
- Для обучения были выбраны модели [YOLO11s](#) и [YOLO12s](#).
- Исходя из полученных метрик, а также качества предсказаний на тестовых изображениях, была выбрана модель для инференса YOLO11s с best.pt весами.
- Было написано приложение [Traffic Sign Detection With YOLO](#) с помощью библиотеки [Gradio](#) и загружено на сайт [Hugging Face](#).
- Полный код итоговой работы, включающий в себя загрузку и предобработку данных, обучение модели находится в файле [RussianTrafficSignProject.ipynb](#)



Планы по доработке

- Добавление *GAN (Generative Adversarial Network)* для генерации синтетических изображений дорожных знаков для решения проблемы дисбаланса данных в категориях.
- Использование других моделей для обучения (*RetinaNet + Focal Loss, Faster R-CNN*)
- Дообучить нашу кастомную модель *YOLO11s("best.pt")* на новых данных (например GTSRB - German Traffic Sign Recognition Benchmark).
- Использование новых аугментаций, увеличение количества эпох обучения и уменьшение learning rate и т.д.



ИНСТИТУТ
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
УНИВЕРСИТЕТА ИННОПОЛИС

Спасибо за внимание!

Контакты



+79828166000



Telegram @edgar_nizaev



Сайт

<https://huggingface.co/NizaevEdgar>

