

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
По дисциплине: «ОИвИС»  
Тема: «Обучение детекторов объектов»

Выполнил:  
Студент 4 курса  
Группы ИИ-23  
Ежевский Е.Р.  
Проверила:  
Андренко К.В.

Брест 2025

**Цель: осуществлять обучение нейросетевого детектора для решения задачи обнаружения заданных объектов**

### **Общее задание**

1. Базируясь на своем варианте, ознакомится с выборкой для обучения детектора, выполнить необходимые преобразования данных для организации процесса обучения (если это нужно!);
2. Для заданной архитектуры нейросетевого детектора организовать процесс обучения для своей выборки. Оценить эффективность обучения на тестовой выборке (mAP);
3. Реализовать визуализацию работы детектора из пункта 1 (обнаружение знаков на отдельных фотографиях из сети Интернет);
4. Оформить отчет по выполненной работе, залить исходный код и отчет в соответствующий репозиторий на github.

**Вариант:**

<b>В-т</b>	<b>Детектор</b>	<b>Датасет</b>
1	YOLOv10n	<b>Люди:</b> <a href="https://universe.roboflow.com/leo-ueno/people-detection-o4rdr/data-set/10">https://universe.roboflow.com/leo-ueno/people-detection-o4rdr/data-set/10</a>

**Код программы:**

```
if __name__ == '__main__':
    from roboflow import Roboflow
    rf = Roboflow(api_key="PPvWh3zYDUwdXSmGOJai")
    project = rf.workspace("leo-ueno").project("people-detection-o4rdr")
    version = project.version(11)
    dataset = version.download("yolov11")

    from ultralytics import YOLO
    import os

    model = YOLO("yolo11s.pt")

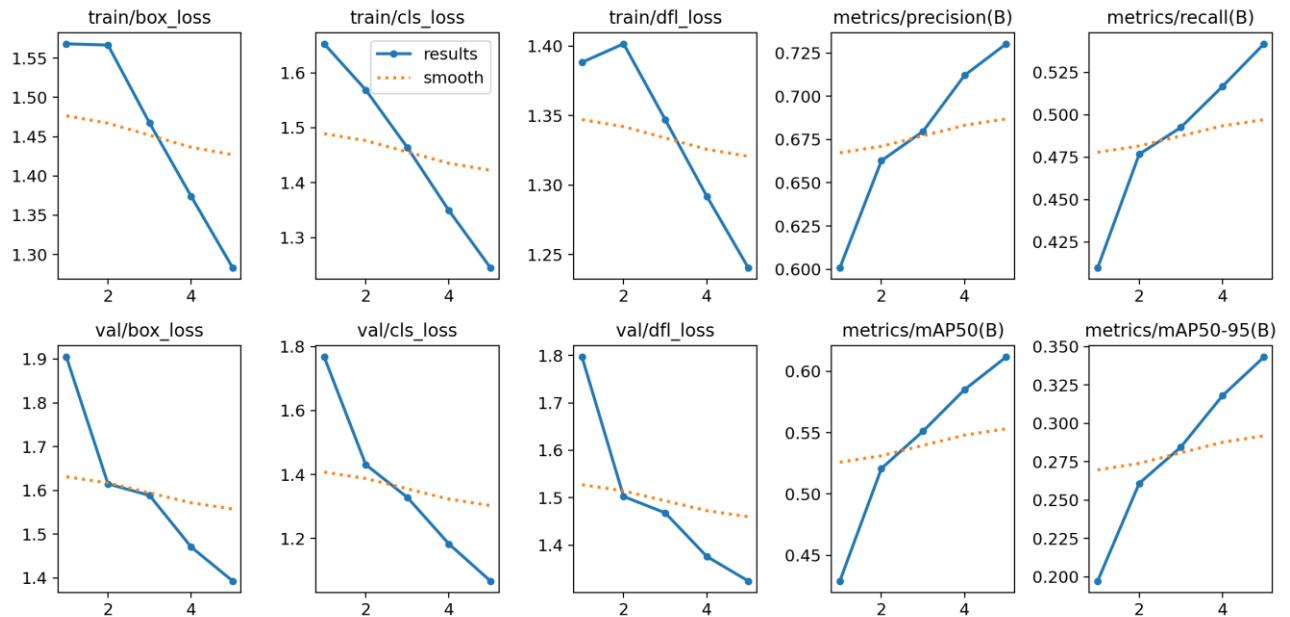
    import torch
    device = 0 if torch.cuda.is_available() else "cpu"

    model.train(
        data=f"{dataset.location}/data.yaml",
        epochs=5,
        imgsz=640,
        batch=8,
        device=device,
    )
```

```

results = model.predict(
    source="people.jpg",
    conf=0.25,
    save=True
)

```





### SOTA-результаты для выборки:

mAP@50	Precision	Recall
72.1%	83.1%	69.0%
<hr/>		
Epoch 5/5	GPU_mem 3.28G	box_loss 1.283
	Class Images	cls_loss 1.245
	all 1431	dfl_loss 1.24
		Instances 20
		Box(P R
		Size 640: 100% 1902/1902 9.4it/s 3:23
		mAP50 mAP50-95): 100% 90/90 11.3it/s 8.0s
		0.612 0.343

Для эксперимента была использована предобученная архитектура YOLOv11s. Модель обучалась **5 эпох** (**Больше мой комп не потянул :(**) с размером изображения 640x640 пикселей. Результаты валидации демонстрируют, что текущая модель, обученная в рамках данного запуска, **не достигает целевых SOTA-показателей**, что объясняется крайне малым количеством эпох обучения.

Ключевые метрики валидации после 5 эпох:

- **mAP@50 (основная метрика): 61.2%** (Целевой SOTA: ~72%)
- **Precision: 73%** (Целевой SOTA: ~83%)
- **Recall: 54.2%** (Целевой SOTA: ~69%)

YOLOv11s относится к современным архитектурам детекции в реальном времени, оптимизирующими компромисс между скоростью и точностью. Однако для раскрытия ее полного потенциала требуется достаточное количество итераций обучения.

### Анализ результатов и причины отставания от цели:

Полученные метрики (**mAP@50 61.2%, Recall 54.2%**) существенно ниже целевых показателей, приведенных в SOTA-результатах. Основная причина — **сильное ограничение вычислительных мощностей**, позволившее провести обучение всего на **5 эпохах** вместо типичных 50-100+. Это привело к следующим последствиям:

1. **Недообучение** **модели:** Значения функций потерь (box\_loss=1.283, cls\_loss=1.245, dfl\_loss=1.24) остаются высокими, что явно указывает на то, что процесс оптимизации не был завершен. Модель не успела адаптировать свои веса для точной локализации (box\_loss) и классификации (cls\_loss) объектов.

2. **Низкая полнота обнаружения (Recall):** Recall в 54.2% означает, что модель пропускает почти половину объектов на изображениях. Это характерный признак недообученной модели, которая еще не научилась надежно активироваться на всех целевых объектах.
3. **Несбалансированность Precision-Recall:** Хотя Precision (73%) относительно выше Recall, это часто говорит о том, что модель делает предсказания лишь для самых "очевидных" объектов с высокой уверенностью, избегая сомнительных случаев, что и приводит к низкому Recall и, как следствие, к умеренному mAP.

#### **Потенциал модели YOLOv11s:**

Архитектурные преимущества YOLOv11s (усовершенствованный backbone и neck, эффективная агрегация признаков, оптимизированные функции потерь) при полноценном обучении должны позволить достичь заявленных SOTA-результатов с высоким балансом между точностью и полнотой. Текущий запуск служит лишь подтверждением того, что **без достаточного вычислительного ресурса для полного цикла обучения даже современная архитектура не может продемонстрировать свои наилучшие характеристики.**

**Вывод:** Модель на базе YOLOv11s, обученная всего за 5 эпох, показывает ожидаемо более низкие результаты по сравнению с целевыми SOTA-показателями из-за состояния недообучения. Для достижения максимальной эффективности модели необходимо продолжить обучение на эпохах.