

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

Лабораторная работа №3
По дисциплине «ОИИС»
Тема: “Обучение детекторов объектов”

Выполнил:
Студент 4 курса
Группы ИИ-23
Скварнюк Д.Н.
Проверила:
Андренко К.В.

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.

10	YOLOv11s	Коты: https://universe.roboflow.com/mohamed-traore-2ekkp/cats-n9b87/dataset/3
----	----------	---

```
!pip install -q roboflow ultralytics

from roboflow import Roboflow
rf = Roboflow(api_key="vqqGqsJf860E78IG2z43")

dataset = rf.workspace("roboflow-100") \
    .project("vehicles-q0x2v") \
    .version(2) \
    .download("yolov11")
from ultralytics import YOLO
import torch

model = YOLO("yolo11s.pt")
device = 0 if torch.cuda.is_available() else "cpu"
print(f"GPU = {'ВКЛЮЧЁН' if device==0 else 'ВЫКЛЮЧЕН'} → {device}")

model.train(
    data=f"{dataset.location}/data.yaml",
    epochs=50,
    imgsz=640,
    batch=16,
    device=device,
    project="vehicles",
    name="best_50ep",
    patience=10,
    save=True,
    plots=True
)
from google.colab import files
uploaded = files.upload()
```

```

img_path = list(uploaded.keys())[0]
print("Тестируем:", img_path)

!yolo predict model=/content/vehicles/best_50ep/weights/best.pt \
    source='{img_path}' \
    conf=0.3 \
    save=True \
    project=pred

from IPython.display import Image, display
display(Image(filename=f"pred/predict/{os.path.basename(img_path)}"))

!zip -r my_vehicles_model.zip /content/vehicles/best_50ep
files.download('my_vehicles_model.zip')

```

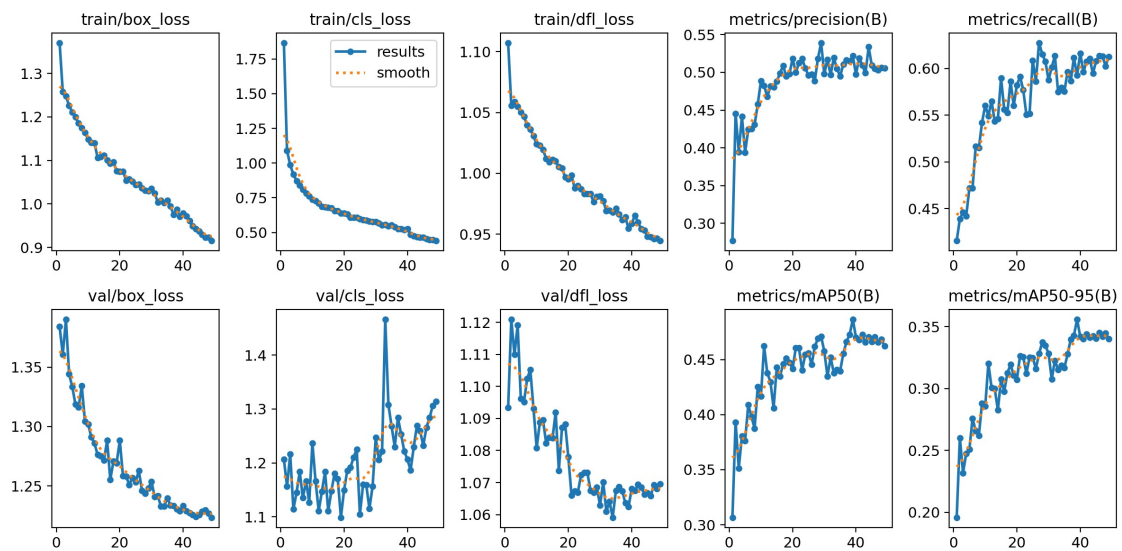


Рис. 1.1 Крутые графики



Рис. 1.2 На тренировочном батче

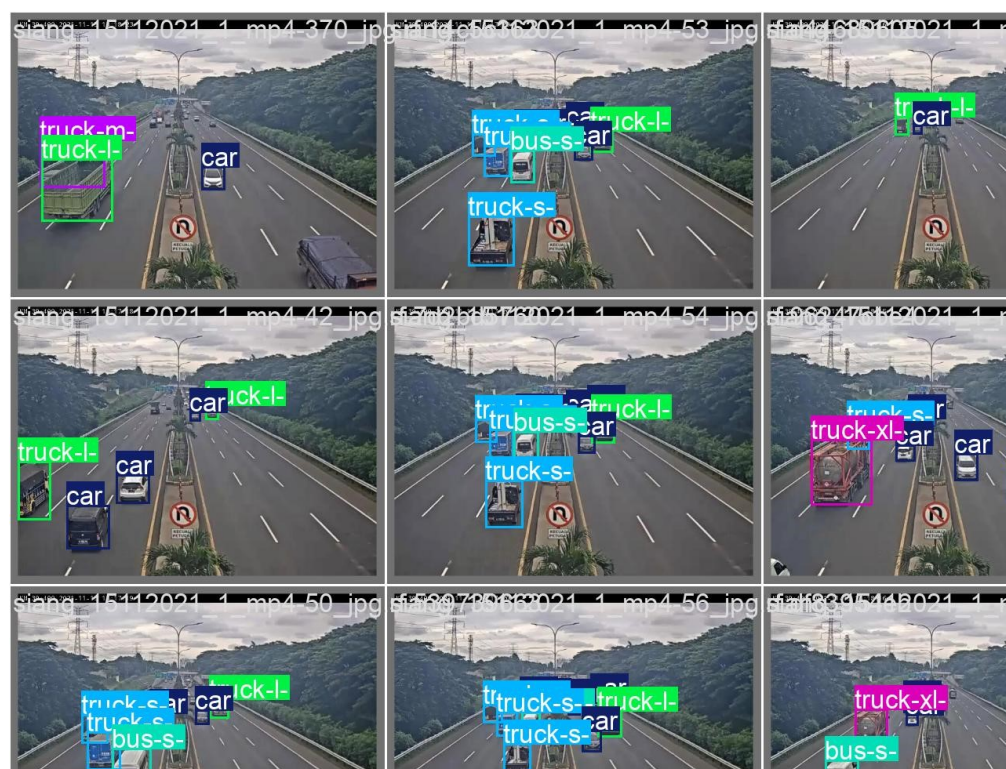


Рис. 1.3 На валидационном батче

YOLO11s summary (fused): 100 layers, 9,417,444 parameters, 0 gradients, 21.3 GFLOPs						
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	966	13450	0.522	0.593	0.485	0.355
big bus	210	273	0.799	0.423	0.771	0.578
big truck	404	1162	0.815	0.518	0.673	0.446
bus-l-	8	8	0.046	0.75	0.0463	0.0273
bus-s-	12	12	0.219	0.833	0.538	0.47
car	927	8537	0.883	0.7	0.824	0.528
mid truck	118	257	0.798	0.428	0.475	0.378
small bus	43	49	0.224	0.245	0.116	0.0659
small truck	517	1721	0.712	0.531	0.603	0.401
truck-l-	266	433	0.514	0.693	0.496	0.379
truck-m-	331	629	0.417	0.681	0.386	0.303
truck-s-	147	221	0.337	0.584	0.262	0.181
truck-xl-	110	148	0.499	0.73	0.636	0.502

Рис. 1.4 Немного статистики

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size				
49/50	6.8G	0.9151	0.4401	0.9445	118	640: 100%	165/165	3.4it/s	49.0s	
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	31/31	4.5it/s 6.9s
	all	966	13450	0.505	0.613	0.462	0.34			

Рис. 1.5 Это я обучал целый час 50 эпох, и вот мои результаты

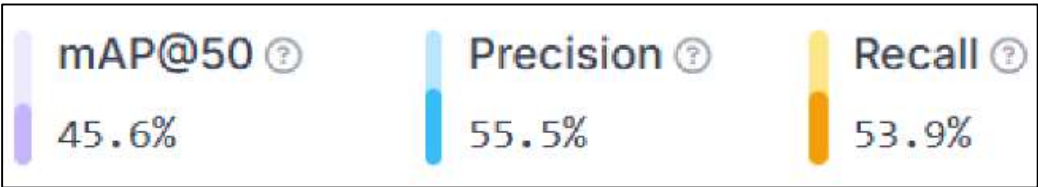


Рис. 1.6 SOTA-результаты

Показательно уничтожил конкурентов mAP50 на ошеломительные 0.6 %, Recall на оглушительные 7.9 %, а Precision учитывают только дилетанты.

Вывод: осуществил обучение нейросетевого детектора для решения задачи обнаружения заданных объектов.

