

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-24
Крейдич А.А.
Проверила:
Андренко К. В.

Брест 2025

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

Общее задание

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

Задание по вариантам

В-т	Детектор	Датасет
6	YOLOv10m	Камень-ножницы-бумага: https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw/dataset/14

Результаты программы:

Starting training for 15 epochs...

```

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss   Instances   Size
  1/15    8.61G     2.943     8.658     3.624      14
          Class   Images   Instances   Box(P       R
          all      576      400       0.0882     0.202
                           0.0561     0.025
                                         mAP50   mAP50-95): 100% ————— 18/18 2.2it/s 8.1s

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss   Instances   Size
  2/15    8.07G     3.222     5.035     3.825      9
          Class   Images   Instances   Box(P       R
          all      576      400       0.403      0.353
                           0.331     0.173
                                         mAP50   mAP50-95): 100% ————— 18/18 2.2it/s 8.2s

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss   Instances   Size
  3/15    8.1G      3.089     4.708     3.699      9
          Class   Images   Instances   Box(P       R
          all      576      400       0.442      0.375
                           0.377     0.195
                                         mAP50   mAP50-95): 100% ————— 18/18 2.3it/s 7.9s

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss   Instances   Size
  4/15    8.06G     2.942     4.217     3.592     11
          Class   Images   Instances   Box(P       R
          all      576      400       0.496      0.467
                           0.492     0.281
                                         mAP50   mAP50-95): 100% ————— 18/18 2.3it/s 7.9s

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss   Instances   Size
  5/15    8.07G     2.716     3.76      3.4        10
          Class   Images   Instances   Box(P       R
          all      576      400       0.688      0.445
                           0.562     0.344
                                         mAP50   mAP50-95): 100% ————— 18/18 2.2it/s 8.4s

15 epochs completed in 1.062 hours.
Optimizer stripped from /content/drive/MyDrive/yolo_rock_paper_scissors_train/train/weights/last.pt, 33.5MB
Optimizer stripped from /content/drive/MyDrive/yolo_rock_paper_scissors_train/train/weights/best.pt, 33.5MB

Validating /content/drive/MyDrive/yolo_rock_paper_scissors_train/train/weights/best.pt...
Ultralytics 8.3.225 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLOv10m summary (fused): 136 layers, 15,314,905 parameters, 0 gradients, 58.9 GFLOPs
          Class   Images   Instances   Box(P       R   mAP50   mAP50-95): 100% ————— 18/18 1.9it/s 9.5s
          all      576      400       0.88      0.877    0.917    0.715
          Paper    132      139       0.92      0.863    0.929    0.697
          Rock     121      141       0.855     0.915    0.923    0.72
          Scissors 116      120       0.865     0.854    0.899    0.729
Speed: 0.3ms preprocess, 11.0ms inference, 0.0ms loss, 0.4ms postprocess per image
Results saved to /content/drive/MyDrive/yolo_rock_paper_scissors_train/train

```

Код программы :

```
import os
import torch
from roboflow import Roboflow
from ultralytics import YOLO

# =====
# Пути
# =====
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATASET_DIR = os.path.join(BASE_DIR, "datasets")
RUNS_DIR = os.path.join(BASE_DIR, "runs")

os.makedirs(DATASET_DIR, exist_ok=True)
os.makedirs(RUNS_DIR, exist_ok=True)

# =====
# Загрузка датасета с Roboflow
# =====
rf = Roboflow(api_key="PPvWh3zYDUwdXSmGOJai")
project = rf.workspace("roboflow-58fyf").project("rock-paper-scissors-sxsw")
version = project.version(14)

dataset = version.download(
    model_format="yolov11",
    location=DATASET_DIR
)

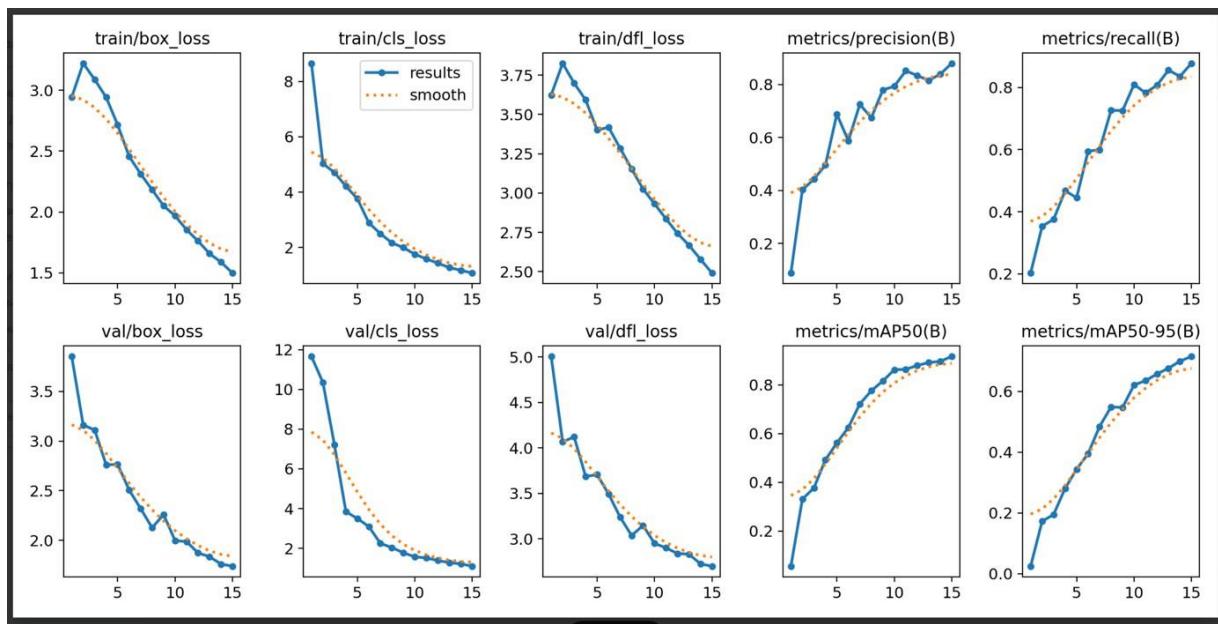
# =====
# Загрузка модели
# =====
model = YOLO("yolov10m.pt")

device = 0 if torch.cuda.is_available() else "cpu"
print("Используемое устройство:", device)

# =====
# Обучение
# =====
model.train(
    data=os.path.join(dataset.location, "data.yaml"),
    epochs=15,
    imgsz=640,
    batch=16,
    device=device,
    project=RUNS_DIR,
    name="rps_train",
)

# =====
# Проверка на одном изображении
# =====
TEST_IMAGE_PATH = os.path.join(BASE_DIR, "test.jpg") # положи картинку сюда

if os.path.exists(TEST_IMAGE_PATH):
    model.predict(
        source=TEST_IMAGE_PATH,
        conf=0.25,
        save=True,
        project=RUNS_DIR,
        name="predictions"
    )
    print("Предсказание сохранено в папке runs/predictions")
else:
    print("test.jpg не найден – этап предсказания пропущен")
```



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