

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

Лабораторная работа №3
По дисциплине: «Обработка изображений в интеллектуальных системах»
Тема: “Обучение детекторов объектов”

Выполнил:
Студент 4 курса
Группы ИИ-24
Капуза Н.А.
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.

4	YOLOv10s	Коты: https://universe.roboflow.com/mohamed-traore-2ekkp/cats-n9b87/dataset/3
---	----------	---

Ход работы:

Код программы:

```
import os
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt
import torch
import gc

# Очистка памяти перед запуском, чтобы удалить "мусор" от прошлых попыток
def clear_gpu_memory():
    torch.cuda.empty_cache()
    gc.collect()

if __name__ == '__main__':
    clear_gpu_memory()

# --- 1. НАСТРОЙКА ПУТИ К ДАТАСЕТУ ---
dataset_yaml_path = os.path.join(os.getcwd(), "Cats-3", "data.yaml")
print(f"Ищем конфиг здесь: {dataset_yaml_path}")

if not os.path.exists(dataset_yaml_path):
    print("⚠️ ВНИМАНИЕ: Файл data.yaml не найден!")
```

```

else:
    print("✓ Файл конфигурации найден.")

# --- 2. ОБУЧЕНИЕ ---
try:
    model = YOLO('yolov10s.pt')
except Exception:
    print("Скачиваем версию v8s (аналог для лабы)...")
    model = YOLO('yolov8s.pt')

# Запускаем обучение с уменьшенным batch
try:
    results = model.train(
        data=dataset_yaml_path,
        epochs=5,
        imgsz=640,

        # --- ВАЖНЫЕ ИЗМЕНЕНИЯ ДЛЯ 4GB VRAM ---
        batch=4,      # Уменьшили с 16 до 4
        workers=2,    # Уменьшили нагрузку на процессор/память
        # -----
        name='cat_yolov10_result_win',
        exist_ok=True # Чтобы перезаписывать папку, если она есть
    )
except RuntimeError as e:
    if "out of memory" in str(e):
        print("\n!!! ВСЁ ЕЩЕ НЕ ХВАТАЕТ ПАМЯТИ !!!")
        print("Попробуйте в коде выше изменить batch=4 на batch=2")
        print("Или уменьшите imgsz=640 на imgsz=416")
    else:
        raise e

# --- 3. ОЦЕНКА (ВАЛИДАЦИЯ) ---
print("\n--- ЗАПУСК ВАЛИДАЦИИ ---")
# Очищаем память перед валидацией
clear_gpu_memory()
metrics = model.val()
print(f'mAP@50: {metrics.box.map50}')
print(f'mAP@50-95: {metrics.box.map}')

# --- 4. ТЕСТ НА СЛУЧАЙНОМ ФОТО ---
print("\n--- ТЕСТ НА ФОТО ИЗ ИНТЕРНЕТА ---")
img_url = "https://cataas.com/cat"

```

```

results = model.predict(source=img_url, conf=0.25, save=True)

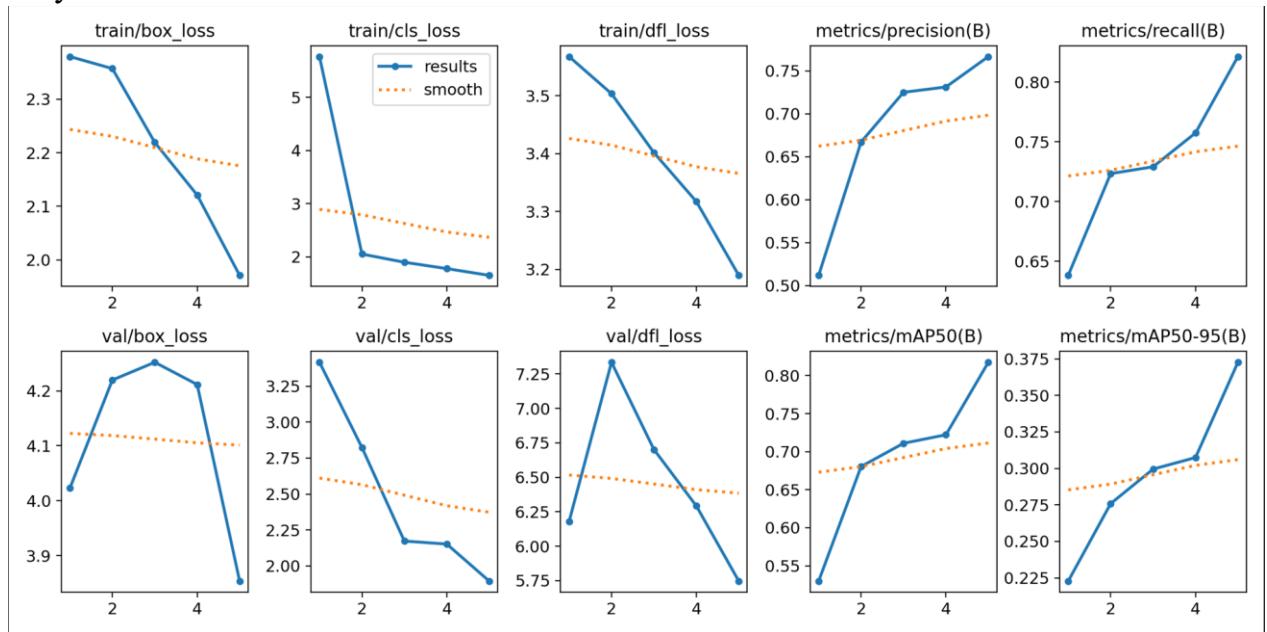
def show_result(result_obj):
    res = result_obj[0]
    im_array = res.plot()
    im_rgb = cv2.cvtColor(im_array, cv2.COLOR_BGR2RGB)

    plt.figure(figsize=(10, 10))
    plt.imshow(im_rgb)
    plt.axis('off')
    plt.title(f"Найдено: {len(res.boxes)}")
    plt.show()

try:
    show_result(results)
except Exception as e:
    print(f"Ошибка при отображении картинки: {e}")

```

Результат:





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