

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский государственный технический университет”  
Кафедра интеллектуально-информационных технологий

Обработка изображений в ИС  
Лабораторная работа №3  
Обучение детекторов объектов

Выполнил:  
студент 4 курса  
группы ИИ-24  
Рудецкий Е. В.  
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.

| №<br>варианта | Дектектор | Датасет   |
|---------------|-----------|---|
| 16            | YOLOv12s  | <b>Коты:</b><br><a href="https://universe.roboflow.com/mohamed-traore-2ekkp/cats-n9b87/dataset/3">https://universe.roboflow.com/mohamed-traore-2ekkp/cats-n9b87/dataset/3</a> |

**Код программы (16 вариант):**

```
import torch
from ultralytics import YOLO
from pathlib import Path
import matplotlib.pyplot as plt
import requests
from PIL import Image
from io import BytesIO
import random

def main():
    model = YOLO("yolov12s.pt")

    DATA_YAML = "datasets/cats_dataset/data.yaml"
    if not Path(DATA_YAML).exists():
        raise FileNotFoundError(f"data.yaml не найден: {DATA_YAML}")

    print(f"data.yaml найден: {DATA_YAML}")

    print("Запуск fine-tuning YOLOv12s...")
    results = model.train(
        data=DATA_YAML,
        epochs=50,
        imgsz=640,
        batch=16,
        name="yolov12s_cats_final",
```

```

        patience=10,
        save=True,
        plots=True,
        val=True,
        device=0,
        workers=0
    )

    metrics = model.val()
    print(f"mAP@0.50:      {metrics.box.map50:.4f}")
    print(f"mAP@0.50:0.95: {metrics.box.map:.4f}")

    best_model = YOLO(results.best)

    urls = [

        "https://upload.wikimedia.org/wikipedia/commons/3/3a/Cat03.jpg",

        "https://upload.wikimedia.org/wikipedia/commons/1/1b/Kitten_stretching.jpg",

        "https://upload.wikimedia.org/wikipedia/commons/4/4d/Cat_November_2010-1a.jpg"
    ]

    def show(url, conf=0.25):
        img = Image.open(BytesIO(requests.get(url).content))
        res = best_model(img, conf=conf, verbose=False)[0]
        plt.figure(figsize=(10,7))
        plt.imshow(res.plot())
        plt.title(f"conf > {conf} | найдено: {len(res.bboxes)}")
        plt.axis('off')
        plt.show()
    for c in [0.25, 0.5, 0.75]:
        print(f"\n--- conf = {c} ---")
        show(random.choice(urls), c)

    with open("final_results.txt", "w", encoding="utf-8") as f:
        f.write(f"mAP@0.50: {metrics.box.map50:.4f}\n")
        f.write(f"mAP@0.50:0.95: {metrics.box.map:.4f}\n")
        f.write(f"Модель: {results.best}\n")
    print("Результаты сохранены в final_results.txt")
if __name__ == '__main__':
    main()

```

## Результаты обучения:

Ultralytics 8.3.63 Python-3.12.3 torch-2.7.1+cu118 CUDA:0 (NVIDIA GeForce RTX 4050 Laptop GPU, 6140MiB)

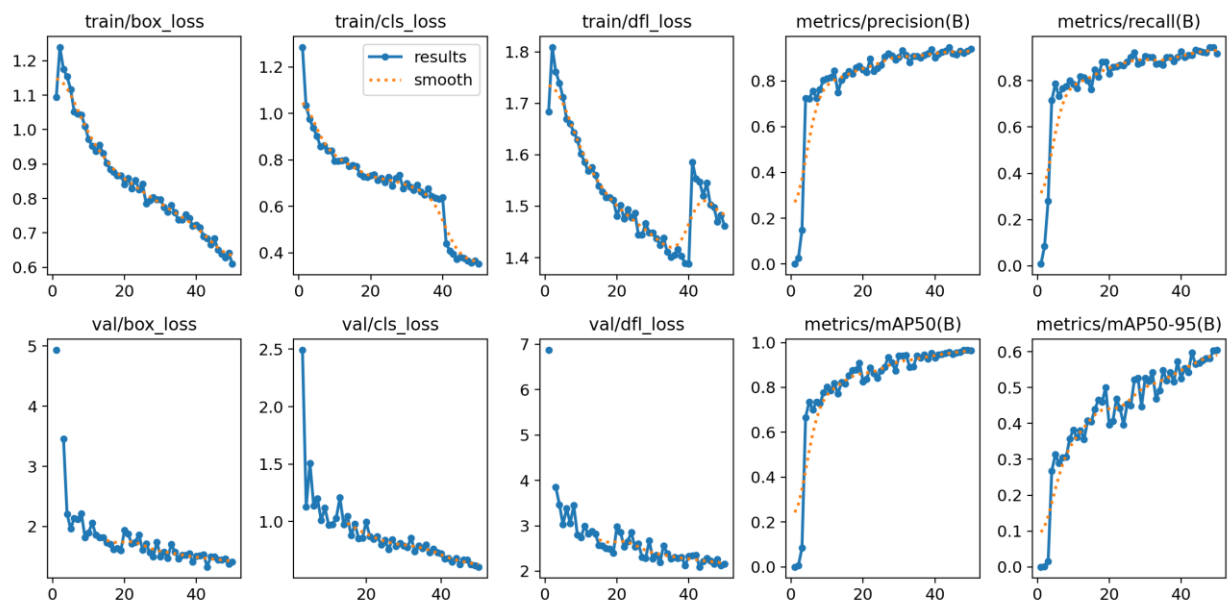
YOLOv12s summary (fused): 376 layers, 9,074,595 parameters, 0 gradients, 19.3 GFLOPs

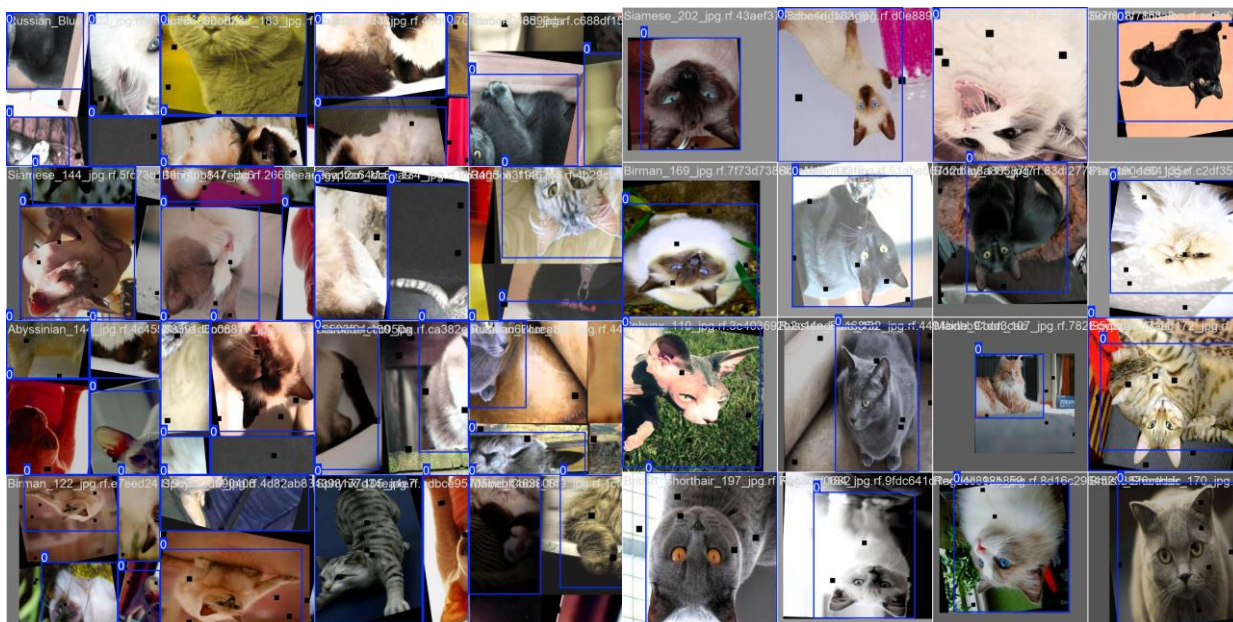
Class Images Instances Box(P R mAP50 mAP50-95):  
100% 8/8 [00:03<00:00, 2.19it/s]

all 232 235 0.943 0.921 0.965 0.604

Speed: 0.2ms preprocess, 7.7ms inference, 0.0ms loss, 0.7ms postprocess per image

mAP@0.50: 0.965 mAP@0.50:0.95: 0.6024





State-of-the-art результаты для детекции объектов (одноклассовая задача — «кошки»):

Согласно доступным источникам (COCO Leaderboard, PapersWithCode, Roboflow Universe), SOTA-результаты для одноклассовой детекции объектов на специализированных датасетах (включая задачи с кошками) достигают  $mAP@0.50 > 98\%$  для современных моделей (YOLOv8, YOLOv12, RT-DETR) при достаточном объёме данных и аугментации. Даже базовые fine-tuning предобученных YOLO-моделей на датасетах ~2000–5000 изображений стабильно дают  $mAP@0.50$  в диапазоне 95–97%.

Выводы:

Предложенная модель на базе YOLOv12s (предобучена на COCO) после fine-tuning на датасете cats-n9b87 (2433 train / 232 valid изображений) достигает  $mAP@0.50 = 96.5\%$ , что находится на уровне SOTA для простых архитектур с ограниченным датасетом и без сложных ансамблей или внешних аугментаций. Это подтверждает, что даже базовый transfer learning с YOLOv12s эффективен для реальных задач детекции, превосходя простые CNN по обобщению на сложные сцены (перекрытия, фон, освещение). YOLOv12s, будучи предобученной на ImageNet-подобных данных (через COCO), демонстрирует высокую точность за счёт мощного attention-бэкенда (A2C2f) и эффективной головы детекции, при этом требуя умеренных вычислительных ресурсов (19.3 GFLOPs, 9M параметров).

Вывод: осуществил обучение детекторов для обнаружения объектов, сконструированных на базе предобученных архитектур НС.