

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

Лабораторная работа №3  
По дисциплине «Обработка изображений в ИС»  
Тема: «Обучение детекторов объектов»

Выполнил:  
Студент 4 курса  
Группы ИИ-21  
Худик А.А.  
Проверил:  
Крощенко А.А.

Брест 2024

**Цель:** осуществлять обучение НС, сконструированных на базе предобученных архитектур НС.

5

## YOLOv5m

**Код программы:**

```
import cv2
import torch
from pathlib import Path
from utils.dataloaders import LoadImages
from utils.general import non_max_suppression, scale_boxes
from utils.plots import Annotator
from utils.torch_utils import select_device
from models.common import DetectMultiBackend

# Настройка модели
YOLO_PATH = Path("") # Путь к YOLOv5
device = select_device('mps') # Используем GPU, если доступно
weights = "/Users/andrewhudik/Downloads/ip_ai_21/reports/Hudik/lab_3/src/yolov5/yolov5m.pt" # Путь к обученной модели
data = str("/Users/andrewhudik/Downloads/ip_ai_21/reports/Парфеев/3/src/data.yaml")

# Загрузка модели
model = DetectMultiBackend(weights, device=device, data=data)
stride, names, pt = model.stride, model.names, model.pt
img_size = 1280

# Функция для обработки видео
def predict_video(source, output):
    dataset = LoadImages(source, img_size=img_size, stride=stride, auto=pt)
    vid_writer = None

    for path, img, im0s, vid_cap, s in dataset:
        # Преобразование кадра
        img = torch.from_numpy(img).to(device).float() / 255.0
        if img.ndimension() == 3:
            img = img.unsqueeze(0)

        # Предсказание
        pred = model(img)
        pred = non_max_suppression(pred)

        # Подготовка для сохранения видео
        for i, det in enumerate(pred):
            im0 = im0s.copy()
            annotator = Annotator(im0, line_width=2, example=str(names))

            if len(det):
                # Масштабируем координаты и добавляем аннотации
                det[:, :4] = scale_boxes(img.shape[2:], det[:, :4], im0.shape).round()
                for *xyxy, conf, cls in reversed(det):
                    label = f"{names[int(cls)]} ({conf:.2f})"
                    annotator.box_label(xyxy, label, color=(255, 0, 0))

            im0 = annotator.result()

        # Инициализируем writer только один раз
        if vid_writer is None:
            fourcc = cv2.VideoWriter_fourcc(*'mp4v') # Кодек для записи
            fps = vid_cap.get(cv2.CAP_PROP_FPS) if vid_cap else 30 # FPS видео
            w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH)) if vid_cap else im0.shape[1]
            h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) if vid_cap else im0.shape[0]
            vid_writer = cv2.VideoWriter(output, fourcc, fps, (w, h))

        # Записываем кадр в видео
        vid_writer.write(im0)

    if vid_writer:
        vid_writer.release()

# Запуск обработки видео
predict_video('/Users/andrewhudik/Downloads/Brest Day Lab Work 3.mp4', "output_day.mp4")
predict_video("/Users/andrewhudik/Downloads/Brest Night Lab Work 3.mp4", "output_night.mp4")
```

**Вывод:** научился осуществлять обучение детекторов объектов.