

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

Лабораторная работа №4
По дисциплине: «Обработка изображений в интеллектуальных системах»
Тема: «Трекинг множественных объектов»

Выполнил:
Студент 4 курса
Группы ИИ-24
Бузель С. Д.
Проверил:
Андренко К. В.

Брест 2025

Цель работы: исследовать применение алгоритмов трекинга на базе обученной сети-детектора объектов.

Общее задание:

1. Используя сеть-детектор, обученный в ЛР 3, реализовать логику для отслеживания множественных объектов, используя библиотеку Ultralytics YOLO;
2. Применять алгоритмы BoT-Sort и ByteTrack (задействовать соответствующие конфигурационные файлы);
3. Исследовать изменения параметров в конфигурационных файлах и их влияние на качество трекинга;
4. В качестве исходных видеоматериалов для экспериментов использовать видео-ролики из сети (например, из YouTube), содержащие множественные объекты классов из ЛР 3;
5. Оформить отчет по выполненной работе, залить исходный код и отчет в соответствующий репозиторий на github.

Код:

```
from ultralytics import YOLO

if __name__ == '__main__':
    # --- ШАГ 1: ЗАГРУЗКА ОБУЧЕННОЙ МОДЕЛИ ---
    # Указываем путь к файлу best.pt, который мы получили в ЛР №3
    model_path = 'D:/REALOIS/lab3/yolov10s_vehicles_lab3/weights/best.pt'
    model = YOLO(model_path)

    # --- ШАГ 2: ВЫБОР ИСХОДНОГО ВИДЕО ---
    video_path = 'D:/REALOIS/lab4/traffic.mp4'

    # --- ШАГ 3: ЗАПУСК ОТСЛЕЖИВАНИЯ ---
    # Используем метод model.track()
    # tracker='botsort.yaml' или 'bytetrack.yaml' - выбор алгоритма трекинга

    print("Начинаем отслеживание с помощью BoT-SORT...")
    results_botsort = model.track(
        source=video_path,
        tracker='botsort.yaml', # Указываем конфигурационный файл трекера
        conf=0.3,              # Порог уверенности для детекции
        iou=0.5,               # Порог IoU для non-max suppression
        device='cpu',
        show=True,             # Отображать видео в реальном времени (!может тормозить на CPU!)
        save=True              # Сохранить видео с результатами трекинга
    )

    print("\nНачинаем отслеживание с помощью ByteTrack...")
    results_bytetrack = model.track(
```

```

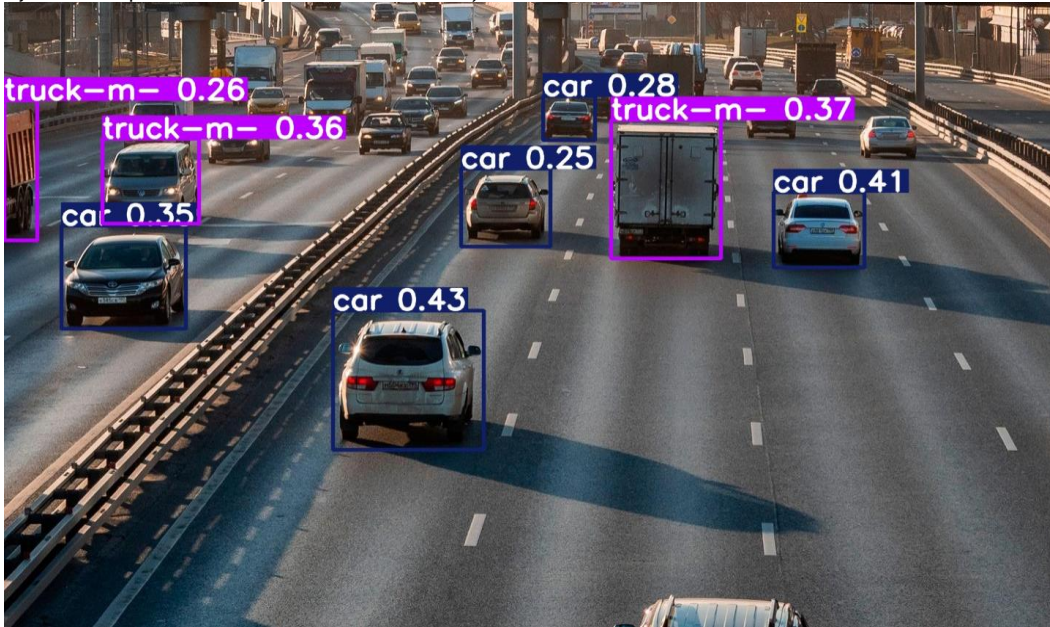
source=video_path,
tracker='bytetrack.yaml', # Меняем трекер на ByteTrack
conf=0.3,
iou=0.5,
device='cpu',
show=True,
save=True
)

```

```

print("\nОбработка завершена.")
print("Результаты сохранены в папку 'runs/detect/track...'")

```



ByteTrack, как правило, лучше работает в плотных сценах с большим количеством перекрытий благодаря своему двухстадийному подходу. ВоT-SORT может быть чуть точнее в плане предсказания траектории за счет фильтра Калмана.

Вывод: в ходе выполнения лабораторной работы исследовал применение алгоритмов трекинга на базе обученной сети-детектора объектов.