

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

Супрунович И. С.

Проверила:

Андренко К. В.

Брест 2025

Цель: 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.

Задание по вариантам

№ в-а	Детектор	Датасет
5	YOLOv10m	Счетчики расхода воды: https://universe.roboflow.com/koer3741-gmail-com/watermetera-mrv2/dataset/1

Код:

```
import cv2  
  
from ultralytics import YOLO  
  
import os
```

```
# ===== КОНФИГУРАЦИЯ =====
```

```
MODEL_PATH = 'best.pt'
```

```
VIDEO_SOURCE = 'test_video.mp4'
```

```
# TRACKER_CONFIG = 'bytetrack.yaml'
```

```
TRACKER_CONFIG = 'botsort.yaml'
```

```
CONF_THRESHOLD = 0.3
```

```
IOU_THRESHOLD = 0.5
```

```
# =====
```

```
def run_tracking():
```

```
    if not os.path.exists(MODEL_PATH):
```

```
        print(f"ОШИБКА: Не найден файл модели по пути: {MODEL_PATH}")
```

```
        print("Сначала выполните ЛР 3 или укажите верный путь.")
```

```
        return
```

```
    if not os.path.exists(VIDEO_SOURCE):
```

```
        print(f"ОШИБКА: Не найден видеофайл: {VIDEO_SOURCE}")
```

```
        print("Пожалуйста, добавьте видеофайл в папку проекта.")
```

```
        return
```

```
    print(f"Загрузка модели: {MODEL_PATH}...")
```

```
    model = YOLO(MODEL_PATH)
```

```
    print(f"Запуск трекинга с алгоритмом: {TRACKER_CONFIG}...")
```

```

cap = cv2.VideoCapture(VIDEO_SOURCE)

w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = int(cap.get(cv2.CAP_PROP_FPS))

output_name =
f"result_{TRACKER_CONFIG.split('.')[0]}_{os.path.basename(VIDEO_SOURCE)}"

out = cv2.VideoWriter(output_name, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))

while cap.isOpened():
    success, frame = cap.read()

    if success:
        results = model.track(
            source=frame,
            persist=True,
            tracker=TRACKER_CONFIG,
            conf=CONF_THRESHOLD,
            iou=IOU_THRESHOLD,
            verbose=False
        )

        annotated_frame = results[0].plot()

        cv2.putText(annotated_frame, f"Tracker: {TRACKER_CONFIG}", (20, 40),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

```

```
cv2.imshow("YOLOv10 Tracking", annotated_frame)
```

```
out.write(annotated_frame)
```

```
if cv2.waitKey(1) & 0xFF == ord("q"):
```

```
    break
```

```
else:
```

```
    break
```

```
cap.release()
```

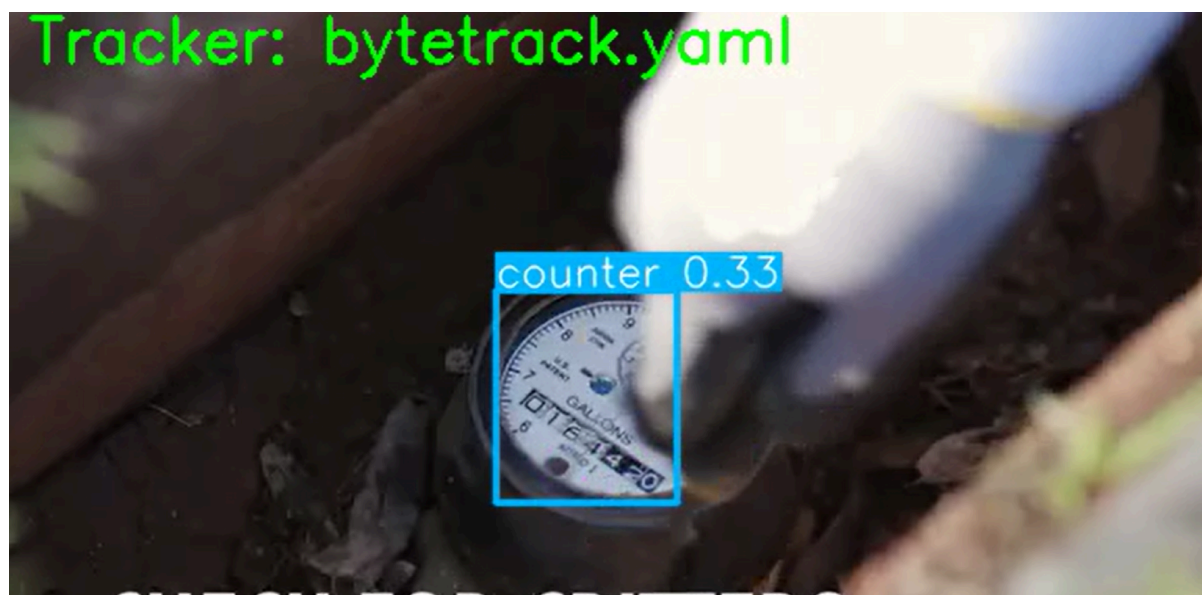
```
out.release()
```

```
cv2.destroyAllWindows()
```

```
print(f'\nГотово! Результат сохранен в файл: {output_name}')
```

```
if __name__ == "__main__":
```

```
    run_tracking()
```



Вывод: осуществлять обучение нейросетевого детектора для решения задачи обнаружения заданных объектов