

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра интеллектуально-информационных технологий

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

Выполнила:
студентка 4 курса
группы ИИ-24
Коцуба Е.М.
Проверила:
Андренко К.В.

Брест 2025

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

Вариант 5

В-т	Детектор	Датасет
5	YOLOv10 m	Счетчики расхода воды: https://universe.roboflow.com/koer3741-gmail-com/watermeteramrv2/dataset/1

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 программы:

```
from roboflow import Roboflow
from ultralytics import YOLO
from google.colab import files
import torch
import os
import glob

device = '0' if torch.cuda.is_available() else 'cpu'
print(f"Используем device: {device} (GPU доступен: {torch.cuda.is_available()})")

print("Загрузите своё видео")
uploaded = files.upload()
video_filename = list(uploaded.keys())[0]
video_path = video_filename
print(f"Видео загружено: {video_path}")

rf = Roboflow(api_key="2OFRWtLUJqdrNuwZhhPk")
project = rf.workspace("koer3741-gmail-com").project("watermeteramrv2")
```

```

dataset = project.version(1).download("yolov8")
data_yaml_path = os.path.join(dataset.location, "data.yaml")

model = YOLO("yolov10m.pt")
model.train(
    data=data_yaml_path,
    epochs=10,
    imgsz=640,
    batch=16 if device != 'cpu' else 8,
    name="yolov10m_watermeter_myvideo",
    device=device,
    patience=10,
    save=True,
    plots=True
)

best_model_path = "runs/detect/yolov10m_watermeter_myvideo/weights/best.pt"
model = YOLO(best_model_path)
print("Классы:", model.names)

def write_tracker_yaml(filename, content):
    with open(filename, "w") as f:
        f.write(content)
    print(f"Создан файл: {filename}")

# ByteTrack custom
write_tracker_yaml("bytetrack_custom.yaml", """tracker_type: bytetrack
track_high_thresh: 0.6
track_low_thresh: 0.1
new_track_thresh: 0.6
match_thresh: 0.85
track_buffer: 30
fuse_score: False
with_reid: False
""")

# BoT-SORT default
write_tracker_yaml("botsort_default.yaml", """tracker_type: botsort
track_high_thresh: 0.5
track_low_thresh: 0.1
new_track_thresh: 0.6
match_thresh: 0.8
track_buffer: 30
fuse_score: True
gmc_method: sparseOptFlow
with_reid: False
""")

# BoT-SORT strict
write_tracker_yaml("botsort_strict.yaml", """tracker_type: botsort
track_high_thresh: 0.6
track_low_thresh: 0.1
new_track_thresh: 0.7
match_thresh: 0.9
track_buffer: 30
fuse_score: True
gmc_method: sparseOptFlow
with_reid: False
""")

# BoT-SORT loose
write_tracker_yaml("botsort_loose.yaml", """tracker_type: botsort
track_high_thresh: 0.4
track_low_thresh: 0.1

```

```

new_track_thresh: 0.5
match_thresh: 0.7
track_buffer: 60
fuse_score: True
gmc_method: sparseOptFlow
with_reid: False
""")

model.track(source=video_path, save=True, tracker="bytetrack.yaml",
name="bytetrack_default")
model.track(source=video_path, save=True, tracker="bytetrack_custom.yaml",
name="bytetrack_custom")
model.track(source=video_path, save=True, tracker="botsort_default.yaml",
name="botsort_default")
model.track(source=video_path, save=True, tracker="botsort_strict.yaml",
name="botsort_strict")
model.track(source=video_path, save=True, tracker="botsort_loose.yaml",
name="botsort_loose")

video_files = glob.glob("runs/track/*/exp*/*.mp4")
video_files.sort()

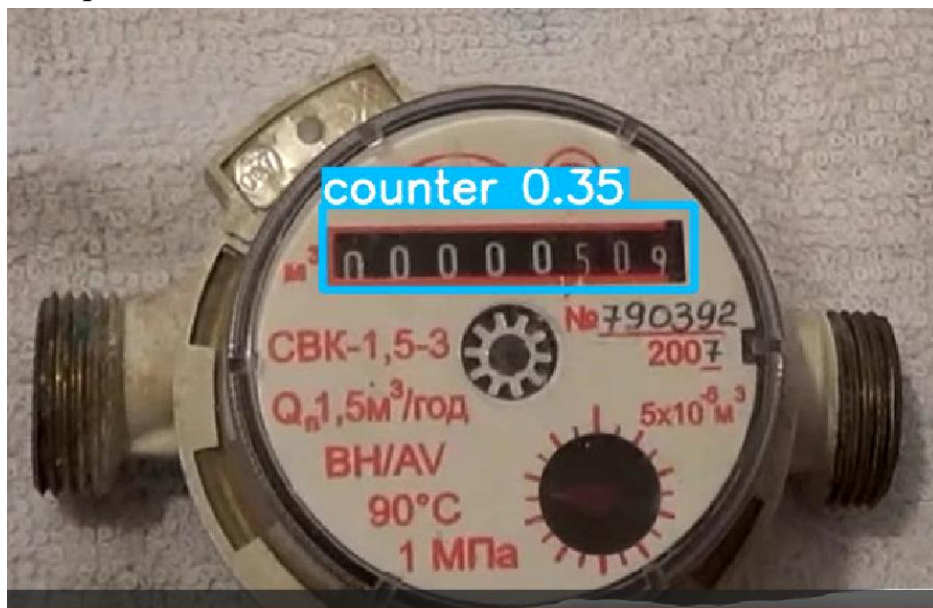
if not video_files:
    print("Видео не найдены! Проверьте, завершился ли трекинг успешно.")
else:
    for i, video in enumerate(video_files, 1):
        print(f"{i}. {video} (размер: {os.path.getsize(video) /
(1024*1024):.1f} MB)")

    print("\nСкачивание видео")
    for video in video_files:
        try:
            files.download(video)
        except Exception as e:
            print(f"Не удалось скачать {video}: {e}")

```

Результат работы программы:

Алгоритм BoT-Sort:



Алгоритм ByteTrack:



Вывод: Реализована логика отслеживания с использованием библиотеки Ultralytics YOLO и трекеров ByteTrack и BoT-Sort. Эксперименты проведены на загруженном видео с изображением счётчиков воды. ByteTrack показал высокую, теряет ID при сильных перекрытиях. BoT-Sort продемонстрировал лучшую стабильность за счёт ReID по внешнему виду, особенно при движении камеры или загромождении объектов.