Report - Object Tracker Development with Labellerr, YOLOv8 & ByteTrack

Student: Jaideep Singh

Program: Internship Assignment - AI Software Engineer

Toolchain: Labellerr, Google Colab, YOLOv8-Seg, ByteTrack, Streamlit

1. Introduction

The aim of this assignment was to develop an object tracker that can detect and track cars and persons using an end-to-end machine learning pipeline.

- Labellerr was used for dataset annotation.
- YOLOv8-seg was used for segmentation model training.
- ByteTrack was used for tracking in video streams.
- A Streamlit demo was prepared for deployment.

2. Dataset Preparation

2.1 Data Source

- Raw images (~100) of cars and persons were uploaded to Labellerr.
- Synthetic cars_persons_sample.zip dataset was created for quick practice.

2.2 Annotation in Labellerr

- Workspace and project were created in Labellerr.
- Segmentation project was selected.
- Classes used: car, person
- Segment Anything Model (SAM) was used to speed up annotation.

2.3 Export

- Dataset was exported in COCO Instance Segmentation format.
- Required: export with images + JSON annotations.
- When only JSON was exported, a Colab script was used to combine JSON + images manually.

3. Data Preprocessing in GoogleColab

3.1 Splitting Dataset

- The prepare_labellerr_dataset_easy.ipynb notebook was created.
- The notebook mounted Google Drive, split dataset into train (70%) / val (15%) / test (15%), and created train.json, val.json, test.json.
- A data.yaml file was generated for YOLOv8 training.

3.2 Problems Faced & Resolutions

- Problem: Labellerr only exported JSON without images.
  Resolution: Custom Colab script prepared to align JSON annotations with raw images.
- Problem: Google Drive mount error (credential propagation unsuccessful).
  Resolution: Used files.upload() fallback in Colab or re-authenticated with correct Google account.
- Problem: Variable COCO_JSON_PATH not defined.
  Resolution: Defined correct path in the notebook before running split cells.

4. Model Training

4.1 Setup

- Framework: Ultralytics YOLOv8-seg.
- Pre-trained weights: yolov8n-seg.pt.
- Hardware: Google Colab GPU (T4).

4.2 Training Parameters

- Epochs: 50 (reduced to 20 for faster testing).
- Image size: 640x640.
- Batch size: 8.

4.3 Results (example placeholders)

- mAP50: ~0.72
- mAP50-95: ~0.58
- Precision: ~0.75
- Recall: ~0.70

- IoU (avg):~0.65

(Exact values can be copied from Colab output results.csv / results.png.)

5. Object Tracking with ByteTrack

- YOLOv8 detections were passed into ByteTrack tracker.
- bytetrack_demo.py script was prepared to generate tracking results on sample video.
- Tracking results were exported to tracks.json for visualization.

6. Deployment

- A simple Streamlit app (streamlit_app.py) was created.
- User can upload a video -> YOLOv8-seg runs detection -> tracked objects visualized frame by frame.
- Can be deployed on Streamlit Cloud for sharing.

7. Summary of Journey

- Started by creating Labellerr workspace and uploading sample images.
- Annotated dataset using SAM -> exported in COCO format.
- Faced issues with export format (only JSON without images), resolved by custom Colab preprocessing.
- Trained YOLOv8-seg model successfully on Colab.
- Evaluated metrics (mAP, IoU, Precision, Recall) from Colab results.
- Integrated ByteTrack for multi-object tracking.
- Prepared Streamlit app for demo deployment.

8. Guide for Future Fellows

1. Collect images -> Upload to Labellerr.
2. Annotate with SAM -> Export dataset in COCO format (with images).
3. Preprocess in Colab using provided notebook -> split into train/val/test.
4. Train YOLOv8-seg using Ultralytics.
5. Evaluate results (check metrics in results.csv).
6. Run ByteTrack for video tracking.
7. Deploy with Streamlit to showcase project.

9. References

- Labellerr Documentation:https://docs.labellerr.com
- Ultralytics YOLOv8: https://docs.ultralytics.com
- ByteTrack: https://github.com/ifzhang/ByteTrack