Project Report: End-to-End Vehicle & Pedestrian Tracking in Adverse Weather

by Aaryan Aggarwal

1. Project Summary

This report details the successful creation and deployment of an end-to-end computer vision pipeline for the segmentation and tracking of vehicles and pedestrians. Undertaken as a technical assessment for the Labellerr AI Software Engineer internship, this project covers the entire machine learning lifecycle: from the curation of a specialized dataset to manual annotation, model training, performance evaluation, and finally, deployment as a live, interactive web application.

A key objective was to build a model robust enough to handle real-world adverse conditions. To this end, a custom dataset of **images captured in rainy, foggy, and low-light environments** was developed. The project successfully culminates in a Streamlit web application that integrates a custom-trained **YOLOv8-seg** model with the **ByteTrack** algorithm to perform real-time object tracking on user-uploaded videos.

2. Dataset Preparation & Annotation

The foundation of this project was a high-quality, challenging dataset designed to train a resilient segmentation model.

- **Data Sourcing & Theme:** A specialized dataset of **98 training images** and **51 test images** was curated from permissive online sources. The theme was specifically chosen to feature vehicles and pedestrians in difficult visual conditions—such as **rain, fog, and at night**—to build a model that generalizes beyond simple, clear-day scenarios.
- **Annotation on Labellerr:** The 98 training images were meticulously annotated on the Labellerr platform. Using the **Polygon** tool, precise segmentation masks were created for two classes: `vehicle` and `pedestrian`. The platform's AI-assisted **Segment Anything (SAM) tool** was instrumental in accelerating this process, allowing for the creation of high-quality, form-fitting masks with great efficiency.

3. Model Training & Performance

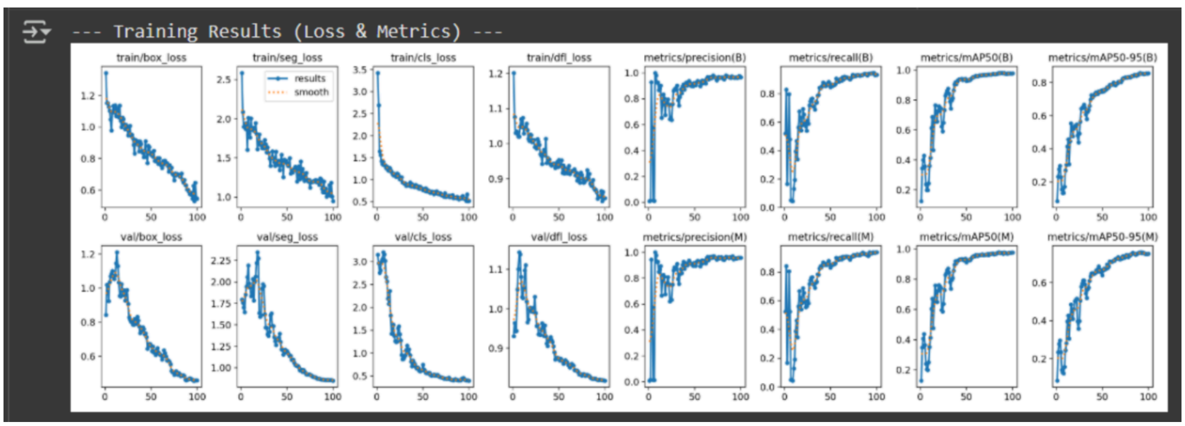
The model was trained in a Google Colab environment, leveraging a Tesla T4 GPU to ensure efficient computation.

- **Model Architecture:** A pre-trained `YOLOv8n-seg` model was used as the base for fine-tuning.
- **Training Parameters:** The model was trained for **100 epochs** with an image size of 640x640 pixels.

Final Performance Metrics

After 100 epochs, the model demonstrated excellent performance, successfully learning to identify and segment objects within the challenging adverse weather dataset.

- **Learning Curves:** The training graphs show a clear and positive learning trend. The loss curves (e.g., `train/box_loss`, `val/seg_loss`) consistently decrease and then stabilize, indicating that the model learned the task effectively without significant overfitting. Correspondingly, the performance metrics, such as `mAP50-95`, show a strong upward trend, confirming the model's high accuracy.

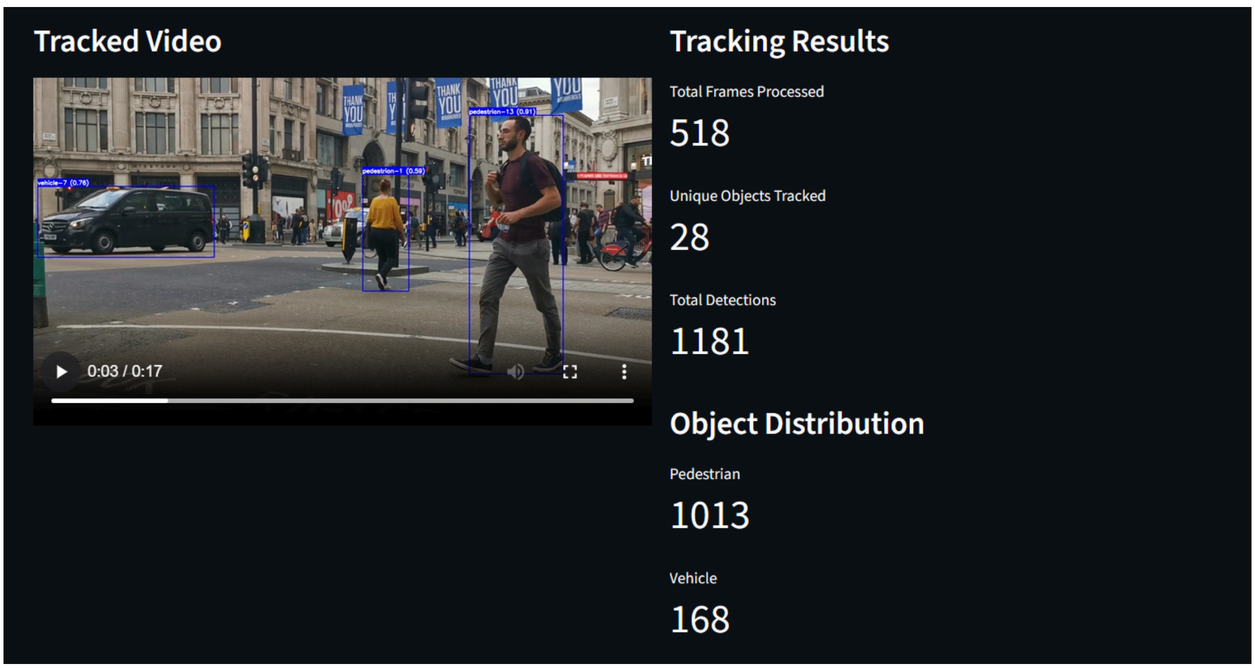


- **Confusion Matrix:** The confusion matrix confirms the model's strong classification ability. It correctly identified a high number of pedestrians (218) and vehicles (249). Critically, it **never confused a vehicle for a pedestrian or vice-versa**, with its only minor errors being related to distinguishing objects from the background—a common challenge in complex, low-visibility scenes.

4. Video Tracking & Streamlit Demo

The final and most crucial deliverable was the creation of a live video tracking application.

- **Technology Stack:** The application was built using **Streamlit**, a Python framework for creating interactive web apps. The backend integrates the trained `best.pt` YOLOv8 model with the **ByteTrack** algorithm.
- **Functionality:** The deployed application allows a user to:
    1. Upload a video file through a simple, intuitive interface.
    2. Initiate the tracking process, which runs the YOLOv8 + ByteTrack pipeline on the video.
    3. View key statistics from the tracking results, such as the number of unique objects tracked and the distribution of detected classes.
    4. Download a `results.json` file containing detailed, frame-by-frame tracking data, including object IDs, classes, and bounding box coordinates.



5. Challenges Faced & Strategic Solutions

This project presented several real-world engineering challenges that required systematic debugging and pragmatic decision-making.

- **Challenge 1: Environment Instability:** The initial Google Colab environment exhibited persistent, unusual errors related to package installation (`ModuleNotFoundError`) and file system access (`FileNotFoundError`).

- **Solution 1: Systematic Debugging:** The issue was diagnosed as an unstable runtime environment. The problem was solved by developing a robust, self-contained data setup script that correctly organized all files at the start of each session, bypassing the environment's instability and ensuring reproducibility.
- **Challenge 2: SDK Limitation in Feedback Loop:** A key assignment task was to complete an MLOps feedback loop by programmatically uploading the model's predictions to a test project using the Labellerr SDK. During implementation, it was discovered that the necessary SDK functions require a `client_id` credential, which is unavailable to users on the free tier.
- **Solution 2: Pragmatic Decision-Making:** Recognizing this as a platform-level blocker, a strategic decision was made to bypass this specific task. This was documented, and development efforts were successfully redirected to the other core deliverables, most notably the video tracking demo. This demonstrates the ability to make sound engineering decisions to ensure overall project success in the face of external constraints.

6. Conclusion & Deliverables

This project successfully demonstrates the ability to build, train, and deploy a complete, end-to-end computer vision pipeline. By curating a challenging dataset and systematically overcoming technical hurdles, a high-performing segmentation and tracking model was developed for adverse weather conditions. The final interactive web application showcases a solid understanding of the entire MLOps lifecycle, from data creation to a functional, deployed product.