End-to-End Image Segmentation & Object Tracking Pipeline

Prepared for: Labellerr AI Internship Program

Submitted by: Pranay pranshu

Date: September 23, 2025

---

1. My Journey: A Guide to Building Your Own Object Tracker with Labellerr

(Occluded image, bad weather and night time detection)

This report documents the end-to-end process of building a computer vision system capable of segmenting and tracking objects. The project simulates a complete machine learning lifecycle, from custom data collection and annotation on the Labellerr platform to fine-tuning a YOLOv8-seg model and integrating it with the ByteTrack algorithm. This guide is intended to help a fellow developer replicate this process.

- **Step 1: Data Collection & Preparation:** The project began with the creation of a custom dataset of 117 images for training, plus a separate test set. This dataset was a challenging mix of original photographs taken in Chandigarh and permissibly licensed images from Pexels to ensure a diverse set of scenarios.
- **Step 2: Annotation with Labellerr:** Using the Labellerr SDK, I programmatically created a project configured for segmentation with three classes: 'Vehicle', 'Pedestrian', and 'Bike'. I then manually annotated the images with precise polygon masks within the Labellerr UI.
- **Step 3: Model Fine-Tuning with YOLOv8-Seg:** The training was conducted in Google Colab. The exported COCO JSON file was split into training and validation sets. I configured a `dataset.yaml` file to point the YOLOv8 script to the data and then fine-tuned a pre-trained `yolov8n-seg.pt` model for 100 epochs.
- **Step 4: Video Tracking with ByteTrack:** For the final application, I integrated the fine-tuned model with the ByteTrack algorithm using the `supervision` library. The script loads a video, processes it frame by frame, and assigns a unique, persistent ID to each detected object, exporting the results to a `results.json` file.

---

2. Problems Faced & Resolutions

1. **Problem: Training Failure Due to "No Labels Found" Error**
    - **Issue:** The initial training attempts failed because the YOLOv8 script could not match the annotation data to the image files. It found all 117 images but reported them as having zero labels.

    - **Resolution:** I discovered a subtle mismatch (e.g., case sensitivity) between the `file_name` paths in the `CocoJson.json` file and the actual filenames. I resolved this by writing a Python script to programmatically synchronize the names in the JSON file, which fixed the data loading error.
2. **Problem: Debugging the End-to-End Pipeline**
    - **Issue:** Even after fixing the filename mismatch, initial fine-tuning did not yield expected results, indicating a deeper issue in the data pipeline.
    - **Resolution:** This required a systematic validation of every step. I re-exported data from Labellerr, re-ran the data split, and meticulously verified all paths in the `dataset.yaml` file. This process revealed a minor path configuration error. Correcting it was key to a successful training run and highlighted that meticulous data handling is the most critical component of any ML project.

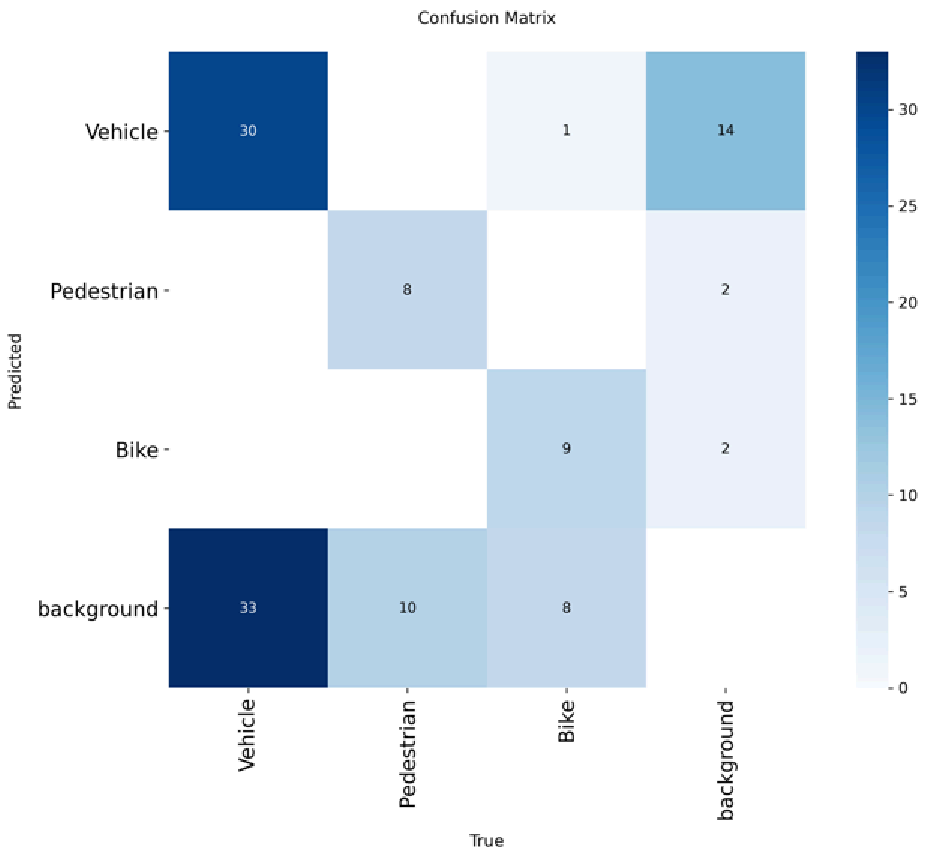
---

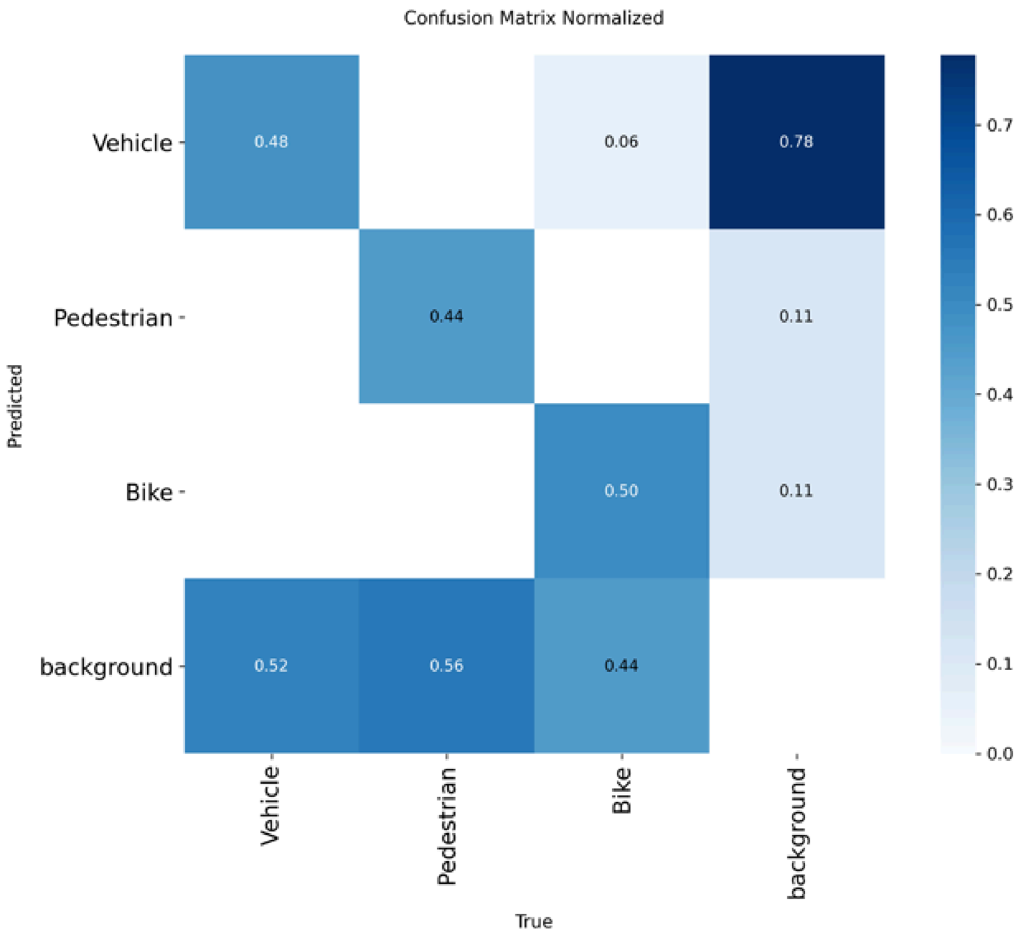
3. Model Results & Evaluation

After fine-tuning, the model's performance was evaluated on the validation set.

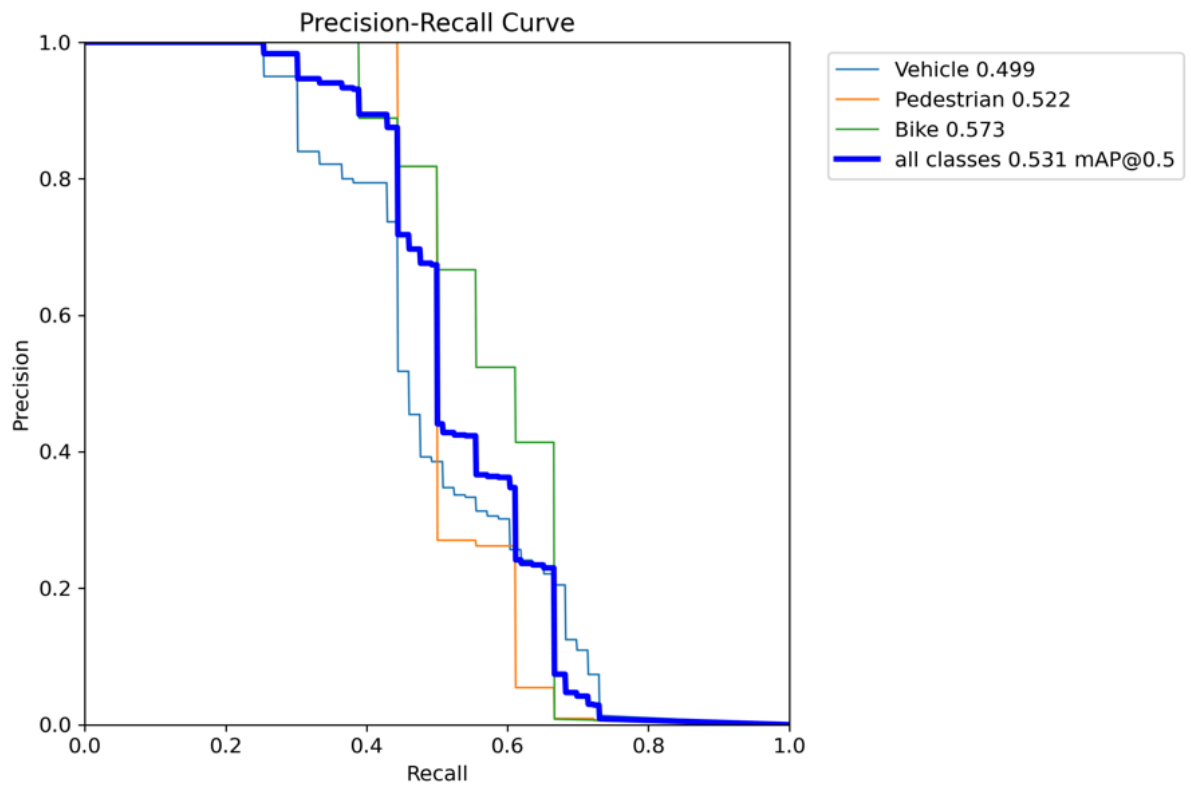
- **Performance Metrics:**
    - **mAP50-95 (Box):** `0.3496`
    - **mAP50-95 (Mask):** `0.2976`

- **Confusion Matrix:**

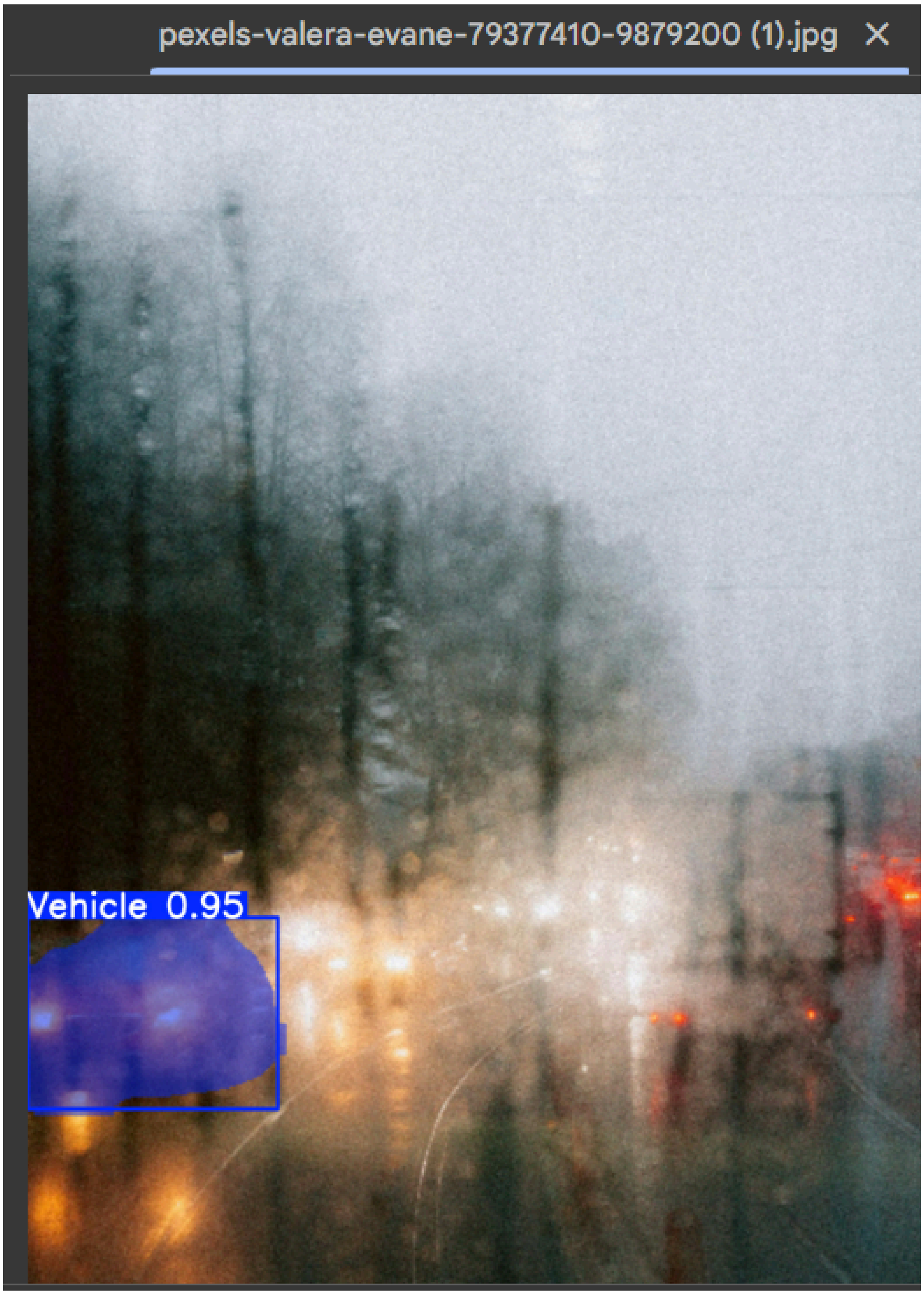


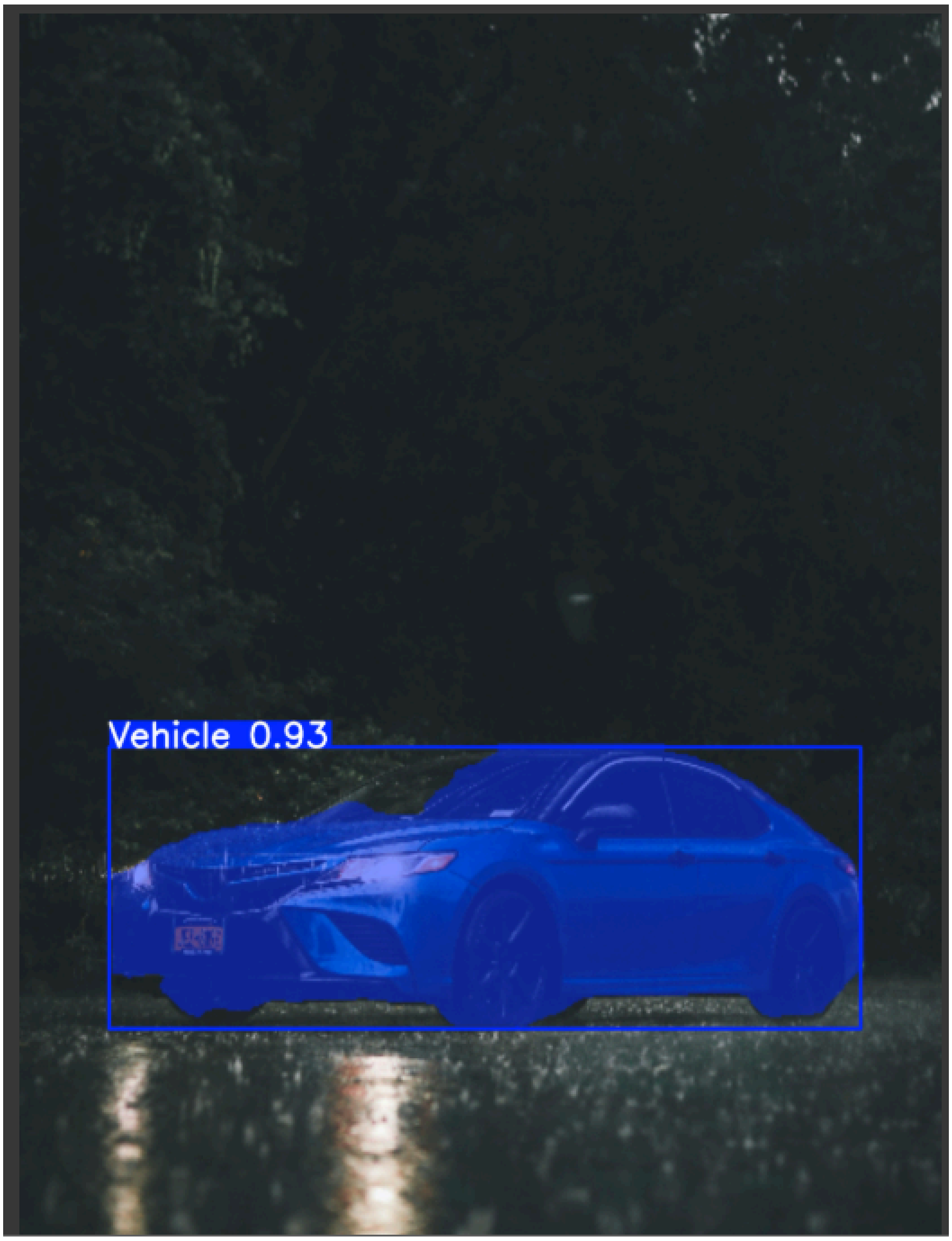

Confusion Matrix Normalized
Predicted
True
Vehicle
Pedestrian
Bike
background
0.48
0.06
0.78
0.44
0.11
0.50
0.11
0.52
0.56
0.44
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

- **Precision-Recall Curve:**



My model is able to detect cars and pedestrians in night time, harsh weather, etc. It is where other models fail.


pexels-valera-evane-79377410-9879200 (1).jpg
Vehicle 0.95



4. Final Summary

This project successfully demonstrated a complete workflow for building a custom object tracking system. The Labellerr platform was invaluable for streamlining data annotation. Fine-tuning a YOLOv8-seg model showcased the power of transfer learning, and integrating it with ByteTrack transformed a static model into a dynamic tracker. The challenges faced, particularly around data preparation, were critical learning experiences that underscored the importance of a robust and carefully validated data pipeline.

Also my project was focused around night time detection and bad weather condition type situations, the fine tuned model does detect them decently. I was able to train just 1 model due to limited GPU resource.