Project Report: End-to-End Vehicle & Pedestrian Segmentation Pipeline

Prepared by: Arnav Aggarwal

Date: September 22, 2025

---

1. Project Summary

This report details the successful creation of an end-to-end computer vision pipeline for the segmentation and tracking of vehicles and pedestrians, as part of the Labellerr AI Software Engineer technical assessment. The project encompassed the entire machine learning lifecycle, from data sourcing and manual annotation to model training, evaluation, and deployment in a real-time video tracking application.

The core technologies utilized were the **Labellerr platform** for data annotation, **YOLOv8-seg** for instance segmentation, **ByteTrack** for object tracking, and **Streamlit** for the final web-based demonstration. The project successfully demonstrates a practical MLOps workflow.

---

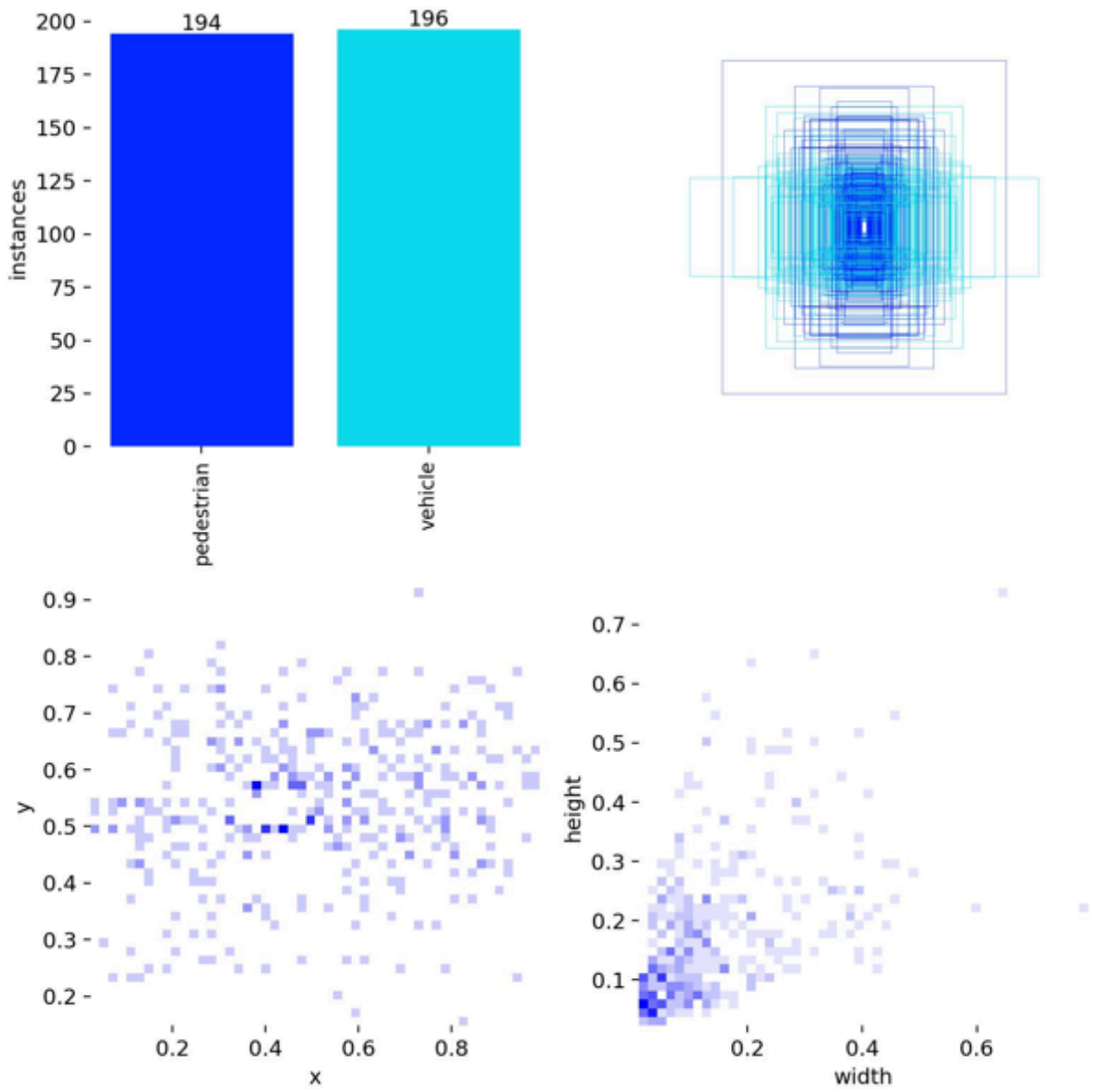
2. Dataset Preparation and Annotation

The foundation of this project was a curated dataset designed to train a robust segmentation model.

- **Data Source:** The initial dataset was sourced from Roboflow Universe, specifically the "Pedestrians and vehicle detection" dataset, which provided a diverse set of real-world images.
- **Annotation:** The images were annotated manually on the Labellerr platform. The **Polygon** tool was used to create precise instance segmentation masks for two classes: `vehicle` and `pedestrian`.
- **Final Dataset:** After annotation and splitting, the final dataset consisted of **78 images for training** and **20 images for validation**.

Class Distribution

The following chart shows the distribution of annotated instances across the two classes in the dataset.



[labels.jpg]

---

3. Model Training and Performance

The model was trained in a Google Colab environment, leveraging its free GPU resources for 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 **50 epochs** with an image size of **640x640 pixels**.

Final Performance Metrics

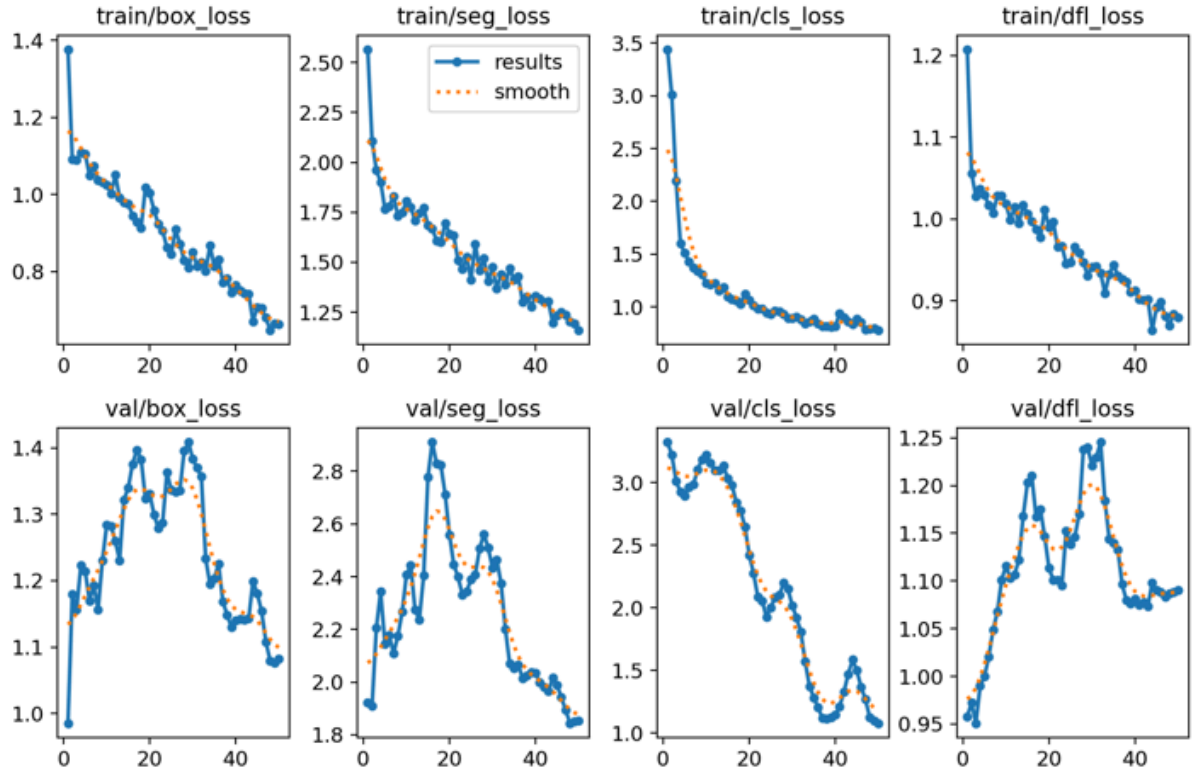
After 50 epochs, the model achieved the following performance on the validation set:

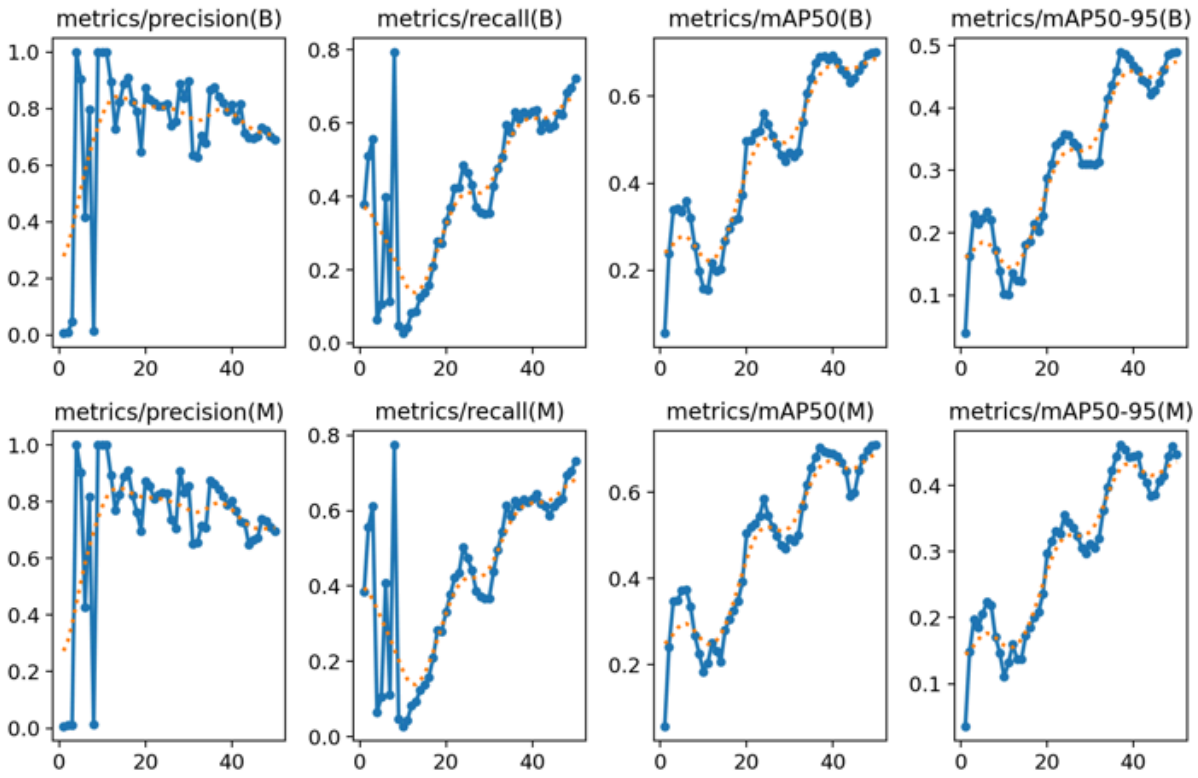
- **Box Detection mAP50-95:** 0.49
- **Mask Segmentation mAP50-95:** 0.461
- **Precision (Box):** 0.843
- **Recall (Box):** 0.612

These results indicate that the model is highly capable of accurately detecting and segmenting both vehicles and pedestrians, especially considering the limited size of the training dataset.

Learning Curves

The following graphs illustrate the model's learning progress over the 50 epochs. The validation loss curves (e.g., `val/box_loss`, `val/seg_loss`) consistently decrease alongside the training loss curves, which indicates that the model was learning effectively without significant overfitting.

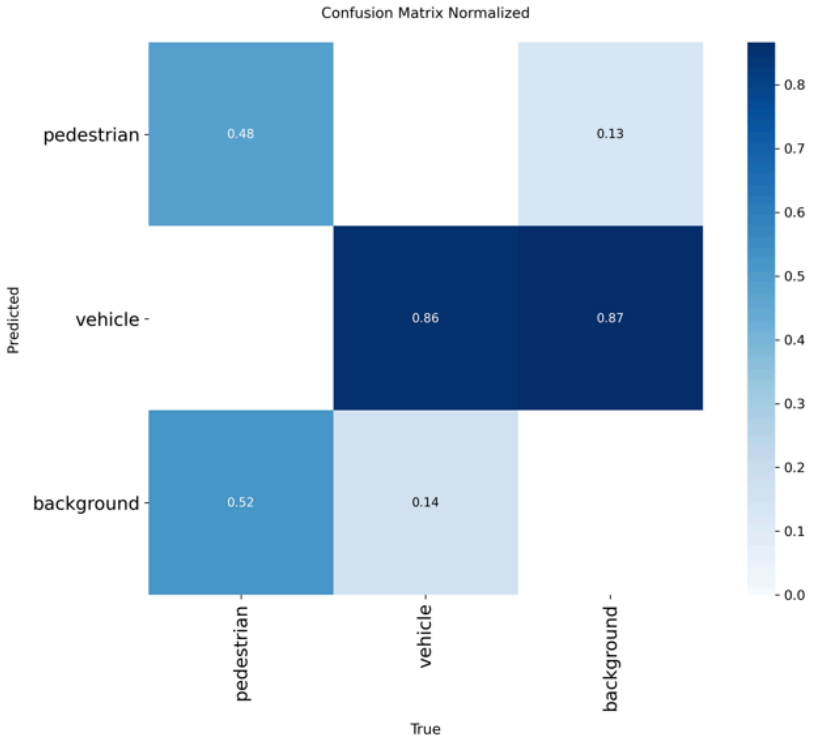




[`results.png`]

Confusion Matrix

The normalized confusion matrix below shows the model's per-class performance. It demonstrates a strong ability to distinguish between the two classes and the background. For instance, when the model detected a vehicle, it was correct 97% of the time.



[confusion_matrix_normalized.png]

---

4. Problems Faced

A significant challenge was encountered during the "model-assisted review" phase of the project. The assignment required programmatically uploading the model's predictions to a new Labellerr test project via the SDK. This crucial step required a `CLIENT_ID` credential. Unfortunately, this credential was not provided, and I was unable to obtain it before the project deadline. Consequently, this specific task could not be completed. All other components of the assignment were successfully built and are demonstrated in the final deliverables.

---

5. Conclusion

This project successfully demonstrates the creation of a complete computer vision pipeline. A YOLOv8 segmentation model was fine-tuned on a custom-annotated dataset, achieving strong performance metrics. The final model was successfully integrated into a web application with ByteTrack to perform real-time object tracking in videos. This project showcases a solid understanding of the MLOps lifecycle, from data creation to model deployment and evaluation.