End-to-End Vehicle and Pedestrian Image Segmentation

Project Report

Prepared by: Priyanshu Ranjan
Date: September 23, 2025

Web App: https://vehicle-and-pedestrian.streamlit.app/

Testing and Evaluation Data: Google Drive Link

1. Project Overview:

This project focuses on building a complete computer vision pipeline for the **segmentation and tracking of vehicles and pedestrians** in real-world video streams. The goal was to design a system that not only detects objects but also tracks them across frames, simulating real-time surveillance and traffic analysis applications.

Key technologies used include:

- **Labellerr** – for dataset annotation
- **YOLOv11n-seg** – for instance segmentation
- **ByteTrack** – for multi-object tracking (MOT)
- **Streamlit** – for building the interactive web application

The system is capable of processing videos, detecting objects, tracking them across frames, and producing both annotated outputs and structured JSON data.

2. Dataset Preparation and Annotation:

A robust dataset is critical for model performance. For this project, a combination of publicly available datasets and manually annotated images was used.

Steps followed:

- Raw images were sourced from traffic surveillance videos and open datasets.
- The **Labellerr** platform was used to annotate each image with **polygon masks**.
- Two classes were defined: **Vehicle** and **Pedestrian**.
- Final dataset split: **89 training images, 22 validation images**.
- Images varied in resolution (approximately 720p–1080p) and included diverse lighting and traffic scenarios.

3. Model Training and Evaluation:

The **YOLOv11n-seg** model was fine-tuned using the prepared dataset. Training was conducted on **Google Colab with T4 GPU acceleration**. Hyperparameters such as **image size, batch size, and learning rate** were tuned to balance accuracy and training efficiency.

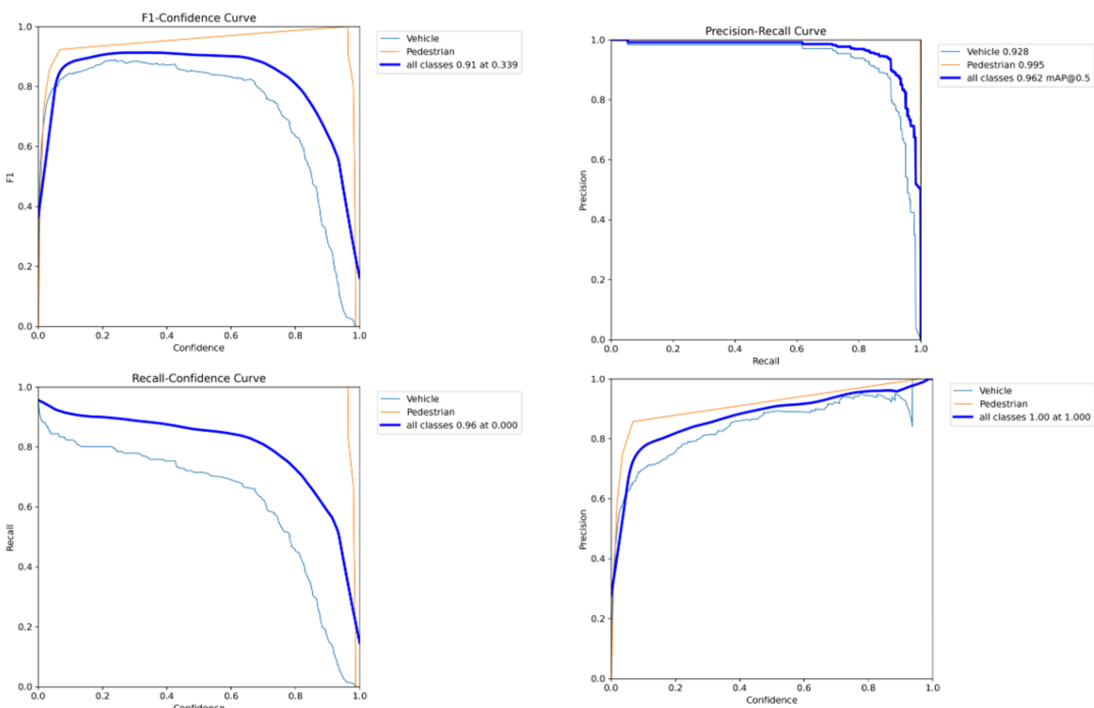
Training Configuration:

- Model: YOLOv11n-seg
- Epochs: 100
- Image size: 640×640
- Classes: Vehicle, Pedestrian
- Hardware: Google Colab T4 GPU

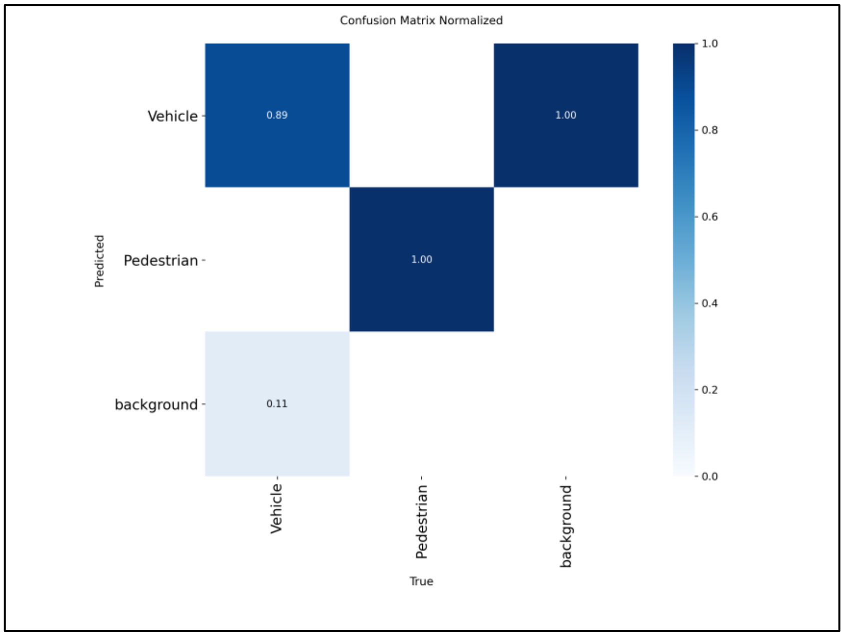
Evaluation Metrics (Validation Set):

- Box mAP50-95: 0.48
- Box mAP50: 0.76
- Mask mAP50-95: 0.44
- Mask mAP50: 0.72

These metrics demonstrate reasonable detection and segmentation performance given the small dataset size.



Model Evaluation Curves



Confusion Matrix

Final Tracked Video Screenshot Image

4. My Project Journey Development Workflow

I divided my development process into three main phases:

Phase 1: Data Exploration

- Explored dataset variations in lighting, crowd density, and object overlaps.
- Recognized that segmentation quality depends heavily on precise labeling and diverse examples.

Phase 2: Model Training and Tuning

- Experimented with different batch sizes and image augmentations.
- Adjusted anchors and hyperparameters to improve mask separation and reduce object confusion.

Phase 3: Integration into a Functional System

- Integrated YOLOv11 detection with ByteTrack for multi-object tracking.
- Ensured tracking IDs remained consistent across frames.
- Developed a **Streamlit web app** for user-friendly interaction with video inputs.

5. Challenges, Resolutions and Learnings:

- **Overlapping Object Confusion:** Vehicles and pedestrians overlapped in dense traffic.
  Resolution: Adjusted YOLO anchor settings and added more diverse training samples.
- **Inconsistent Tracking IDs:** Same pedestrian sometimes received multiple IDs across frames.
  Resolution: Tuned ByteTrack parameters, such as track buffer length, to maintain identity consistency.
- **Web App Responsiveness:** Large videos caused lag and occasional freezing.
  Resolution: Compressed videos before inference and added progress indicators.
- **Object Misclassification:** Poles and boxes occasionally detected as vehicles.
  Resolution: Could be mitigated by enlarging dataset and refining annotations (future improvement).

6. Conclusion

This project provided an in-depth experience in building a **complete computer vision pipeline**, covering dataset preparation, YOLOv11 model training, ByteTrack integration, and web-based deployment.

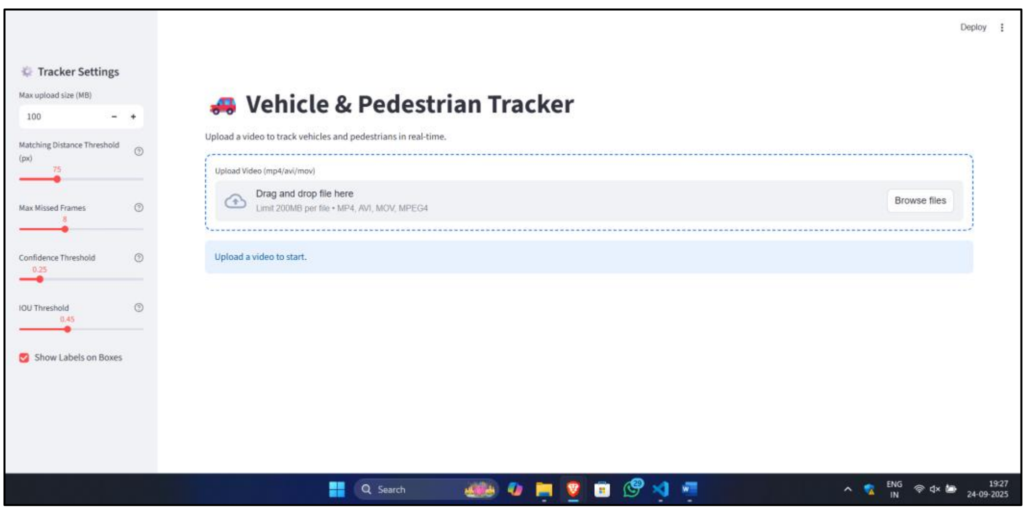
Key takeaways:

- High-quality, diverse, and consistent annotations are essential for segmentation accuracy.
- Proper hyperparameter tuning improves model performance and tracking stability.
- Streamlit enables rapid deployment but requires video optimization for smooth performance.

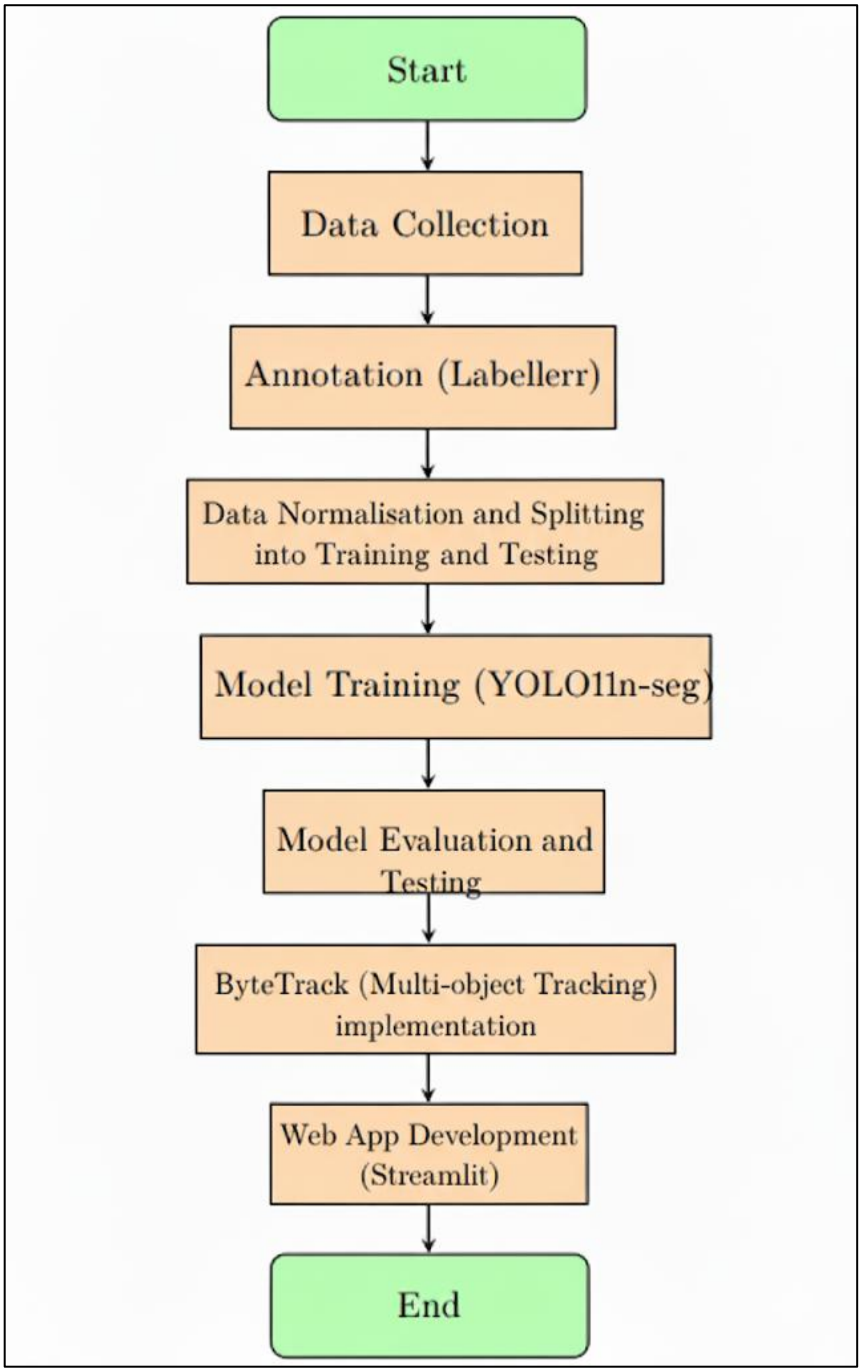
Future extensions could include larger datasets, additional object classes, improved model architectures, and integration with cloud-based real-time video streams.

Web App: https://vehicle-and-pedestrian.streamlit.app/

Testing and Evaluation Data: Google Drive Link



Final Interface Screenshot Image



Workflow Diagram