Vehicle and Pedestrian Detection & Tracking using YOLOv8 + ByteTrack
Prepared by: *Anirudh Sharma*
Date: *2025*

---

1. Introduction

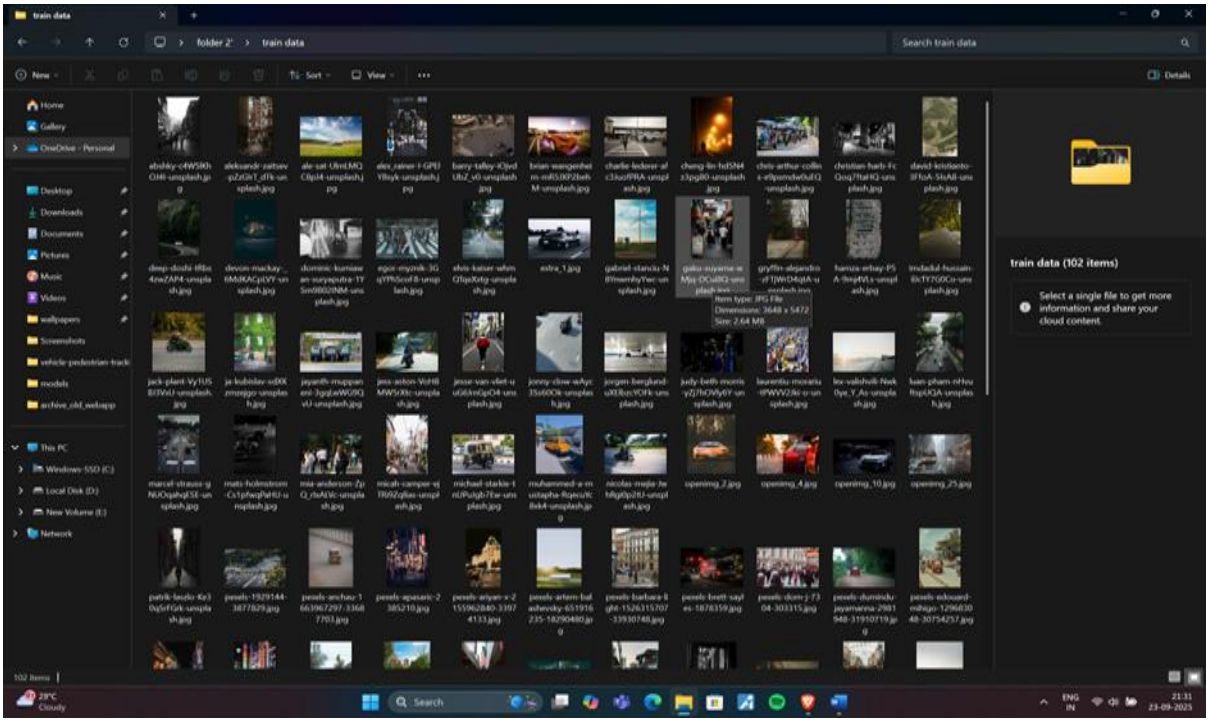
This project demonstrates an end-to-end pipeline for object detection and tracking using **YOLOv8 segmentation** integrated with **ByteTrack**. The objective was to build a model capable of detecting and tracking **vehicles and pedestrians** in videos, and then deploy a user-friendly web demo.

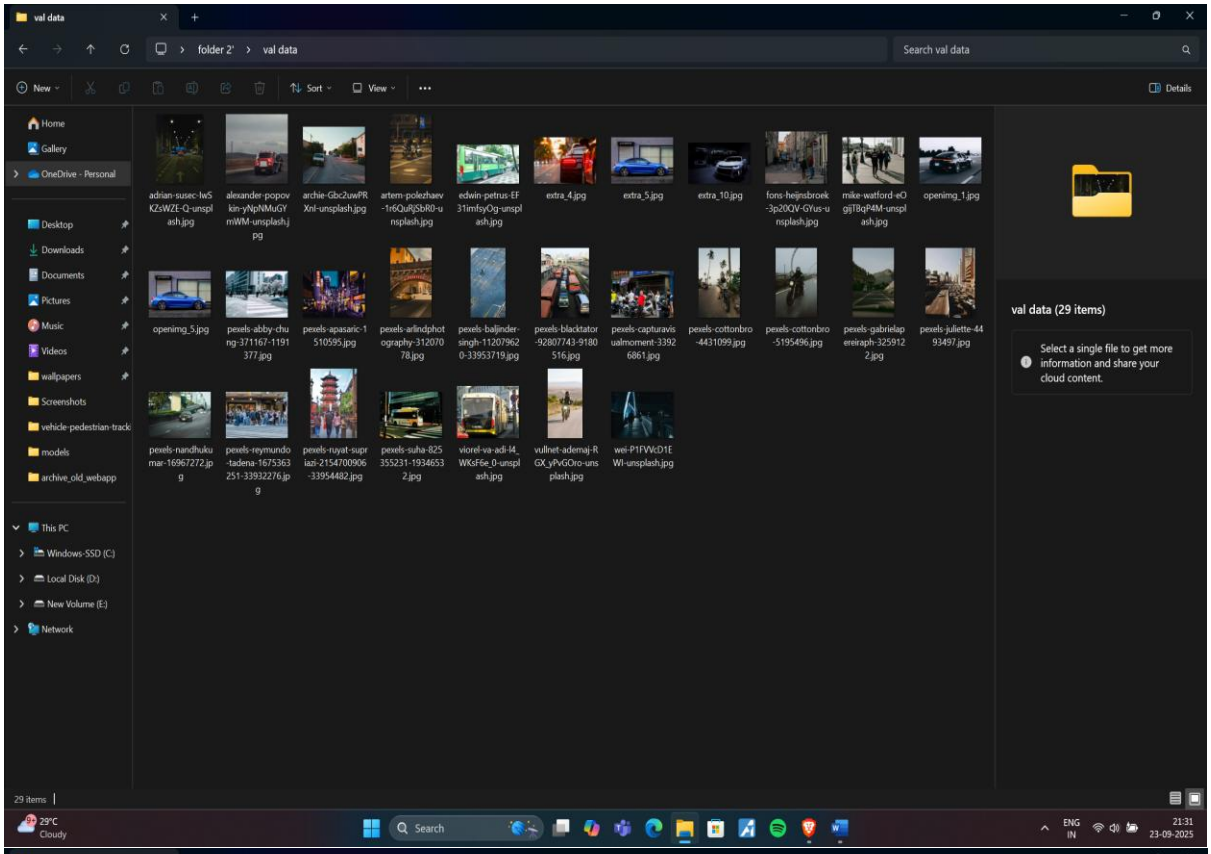
The project was part of the Labellerr PEC AI Software Engineer assignment, and includes dataset preparation, model training, evaluation, web app development, deployment, and documentation.

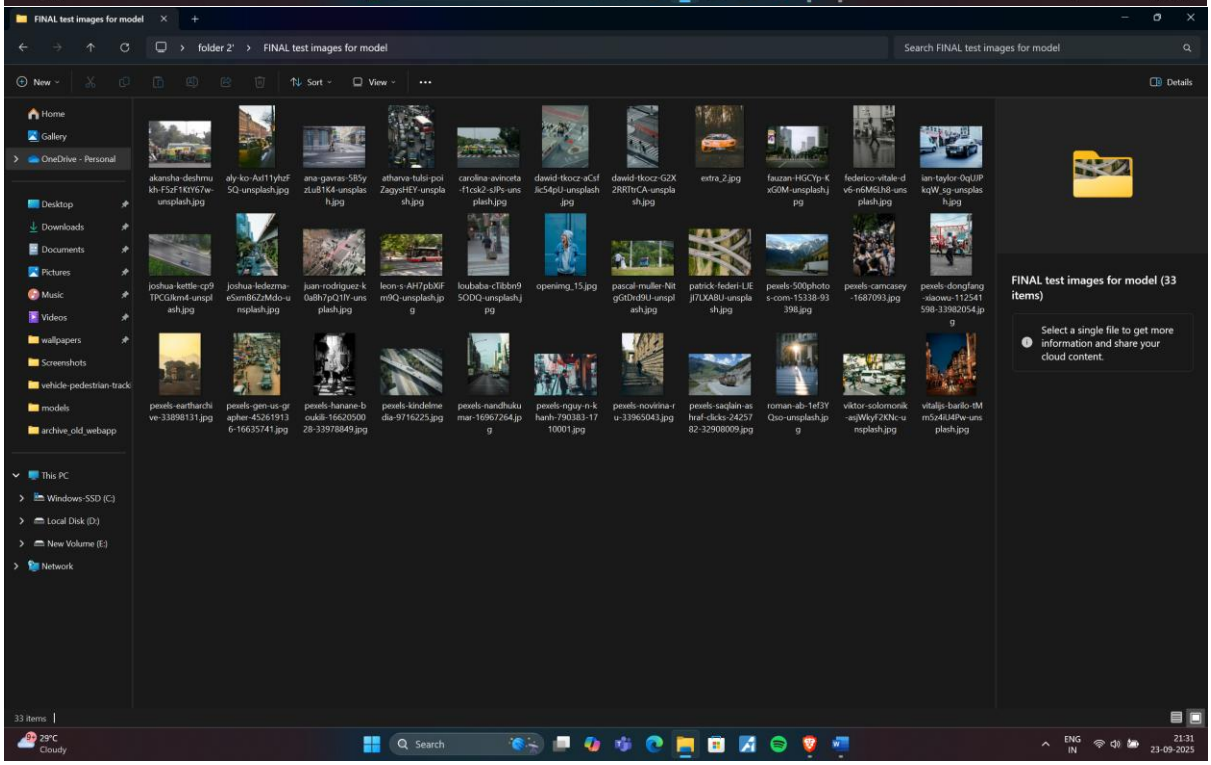
---

2. Dataset Preparation

- **Data Sources:** Unsplash, Pexels, OpenImages.
- **Classes:**
    1. Vehicle
    2. Pedestrian
- **Size:** 183 images split into train/val/test.
- **Annotation Tool:** Labellerr platform.
- **Format:** YOLO (bounding boxes + segmentation masks).

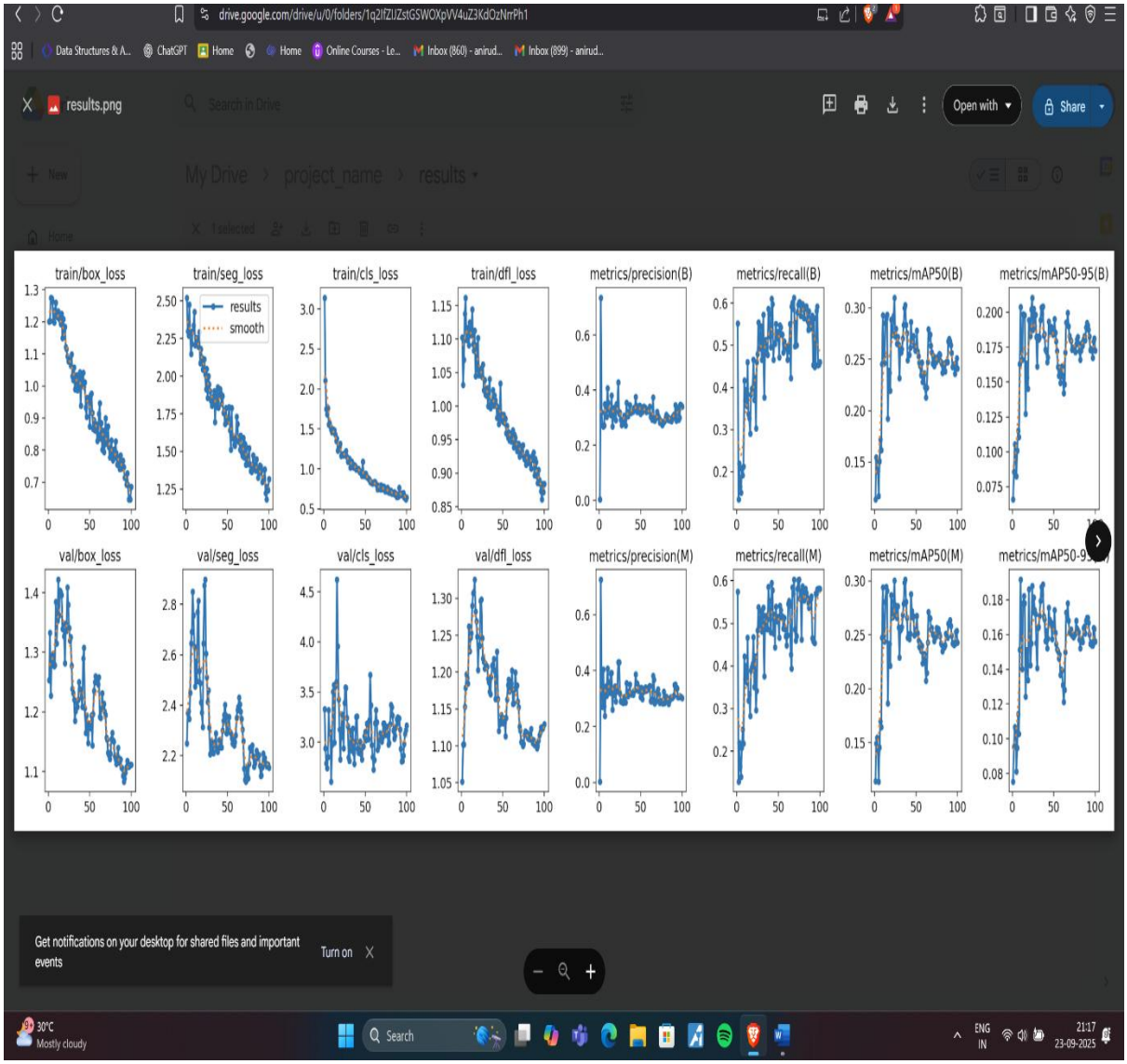


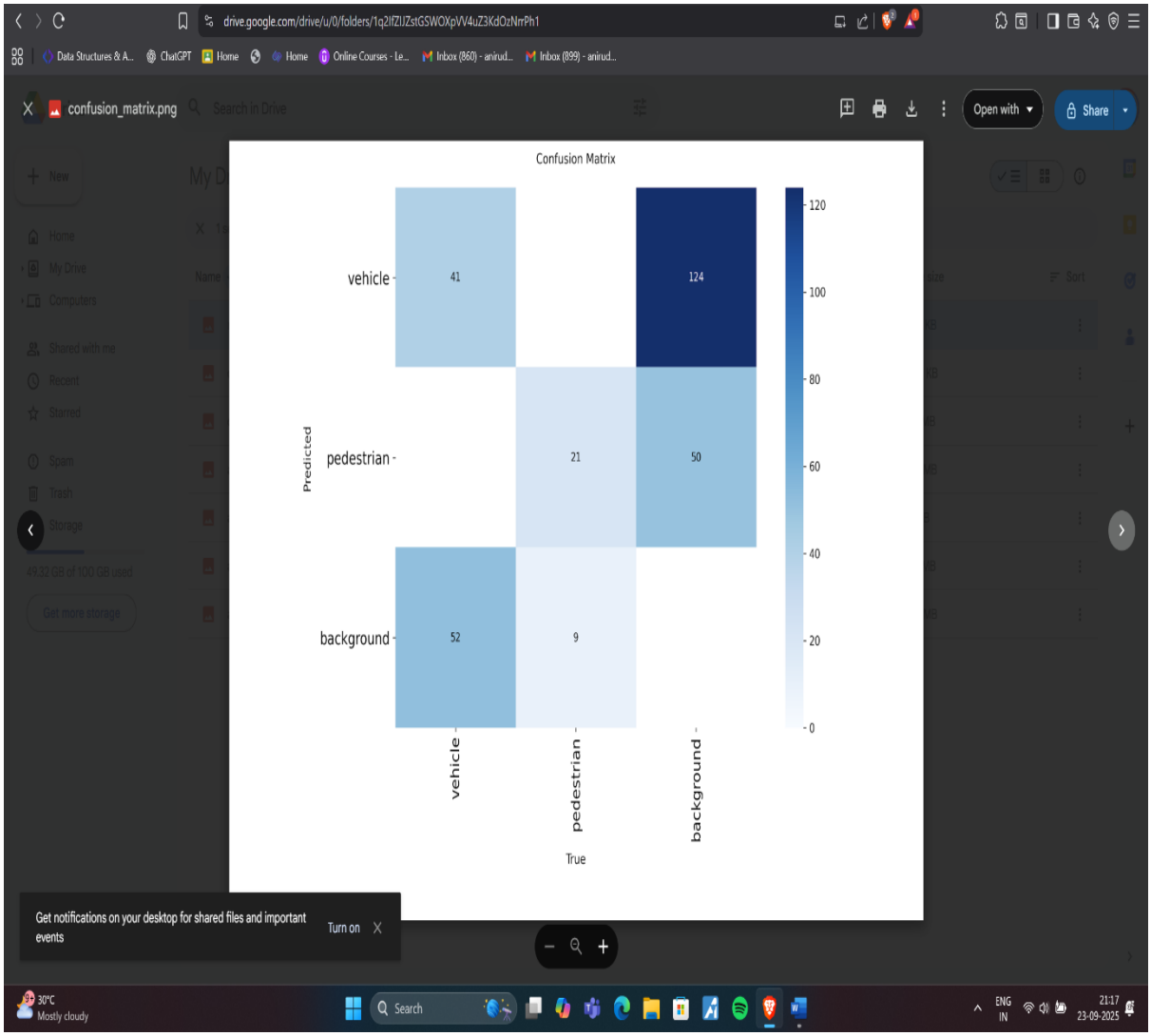

val data
folder 2' › val data
Search val data
New
Sort
View
Details
Home
Gallery
OneDrive - Personal
Desktop
Downloads
Documents
Pictures
Music
Videos
wallpapers
Screenshots
vehicle-pedestrian-track
models
archive_old_webapp
This PC
Windows-SSD (C:)
Local Disk (D:)
New Volume (E:)
Network
adrian-susec-lwSKZsWZE-Q-unsplash.jpg
alexander-popovkin-yNpNMuGYmWM-unsplash.jpg
archie-Gbc2uwPRXnI-unsplash.jpg
artem-polezhaev-1r6QuRjSbR0-unsplash.jpg
edwin-petrus-EF31imfsyOg-unsplash.jpg
extra_4.jpg
extra_5.jpg
extra_10.jpg
fons-heijnsbroek-3p20QV-GYus-unsplash.jpg
mike-watford-eOgjTBqP4M-unsplash.jpg
openimg_1.jpg
openimg_5.jpg
pexels-abby-chung-371167-1191377.jpg
pexels-apasaric-1510595.jpg
pexels-arlindphotography-31207078.jpg
pexels-baljinder-singh-11207962-33953719.jpg
pexels-blacktator-92807743-9180516.jpg
pexels-capturavisualmoment-33926861.jpg
pexels-cottonbro-4431099.jpg
pexels-cottonbro-5195496.jpg
pexels-gabrielapereiraph-3259122.jpg
pexels-juliette-4493497.jpg
pexels-nandhukumar-16967272.jpg
pexels-reymundo-tadena-167536251-33932276.jpg
pexels-ruyat-supriazi-2154700906-33954482.jpg
pexels-suha-825355231-19346532.jpg
viorel-va-adi-I4_WKsF6e_0-unsplash.jpg
vullnet-ademaj-RGX_yPvGOro-unsplash.jpg
wei-P1FVWcD1EWI-unsplash.jpg
val data (29 items)
Select a single file to get more information and share your cloud content.
29 items
29°C
Cloudy
Search
ENG
IN
21:31
23-09-2025


FINAL test images for model
folder 2' › FINAL test images for model
Search FINAL test images for model
New
Sort
View
Details
Home
Gallery
OneDrive - Personal
Desktop
Downloads
Documents
Pictures
Music
Videos
wallpapers
Screenshots
vehicle-pedestrian-track
models
archive_old_webapp
This PC
Windows-SSD (C:)
Local Disk (D:)
New Volume (E:)
Network
akansha-deshmukh-F5zF1KtY67w-unsplash.jpg
aly-ko-Aal11yhzF5Q-unsplash.jpg
ana-gavras-585yzLu81K4-unsplash.jpg
atharva-tulsi-poiZagysHEY-unsplash.jpg
carolina-avinceta-f1csk2-sJPs-unsplash.jpg
dawid-tkocz-aCsfJic54pU-unsplash.jpg
dawid-tkocz-G2X2RRTrCA-unsplash.jpg
extra_2.jpg
fauzan-HGCYp-KxG0M-unsplash.jpg
federico-vitale-dv6-n6M6Uh8-unsplash.jpg
ian-taylor-0qUJPkqW_sg-unsplash.jpg
joshua-kettle-cp9TPCGjkm4-unsplash.jpg
joshua-ledezma-e5xm86ZzMdo-unsplash.jpg
juan-rodriguez-k0a8h7pQ1fY-unsplash.jpg
leon-s-AH7pbXFm9Q-unsplash.jpg
loubaba-cTibbn95ODQ-unsplash.jpg
openimg_15.jpg
pascal-muller-NtgGtDrd9IU-unsplash.jpg
patrick-federi-LJEjI7LXA8U-unsplash.jpg
pexels-500photos-com-15338-93398.jpg
pexels-camcasey-1687093.jpg
pexels-dongfang-xiaowu-112541598-33982054.jpg
pexels-earthharchive-33898131.jpg
pexels-gen-us-grapher-45261913 6-16635741.jpg
pexels-hanane-boukili-166205008 28-33978849.jpg
pexels-kindelmedia-9716225.jpg
pexels-nandhukumar-16967264.jpg
pexels-nguy-n-khanh-790383-1710001.jpg
pexels-novirina-ru-33965043.jpg
pexels-saqlain-ashraf-clicks-24257 82-32908009.jpg
roman-ab-1ef3YQso-unsplash.jpg
viktor-solomonik-asjWkyF2KNc-unsplash.jpg
vitaljs-barilo-tMm5z4iLl4Pw-unsplash.jpg
FINAL test images for model (33 items)
Select a single file to get more information and share your cloud content.
33 items
29°C
Cloudy
Search
ENG
IN
21:31
23-09-2025

3. Model Training

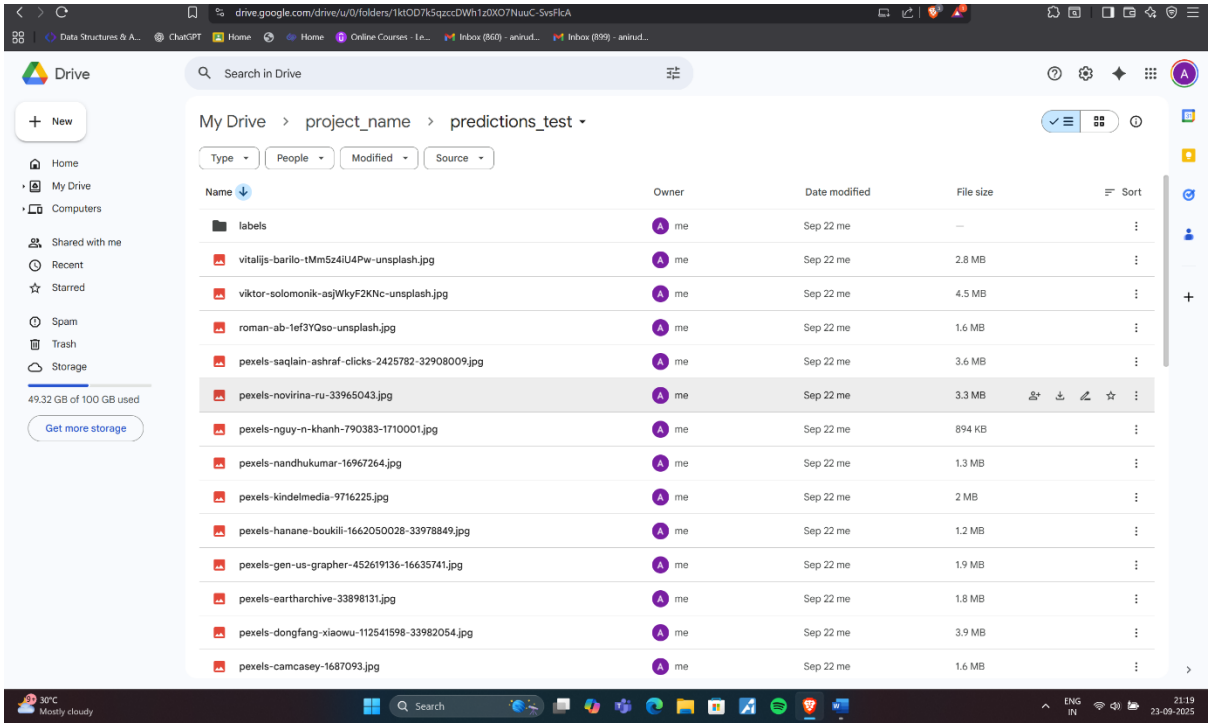
- **Model:** YOLOv8n-Seg (lightweight segmentation).
- **Environment:** Google Colab (Tesla T4 GPU).
- **Parameters:**
    - Epochs: 100
    - Image size: 640
    - Batch size: 8
- **Output:**
    - best.pt → trained weights.
    - Precision-Recall curves, IoU, and mAP metrics.

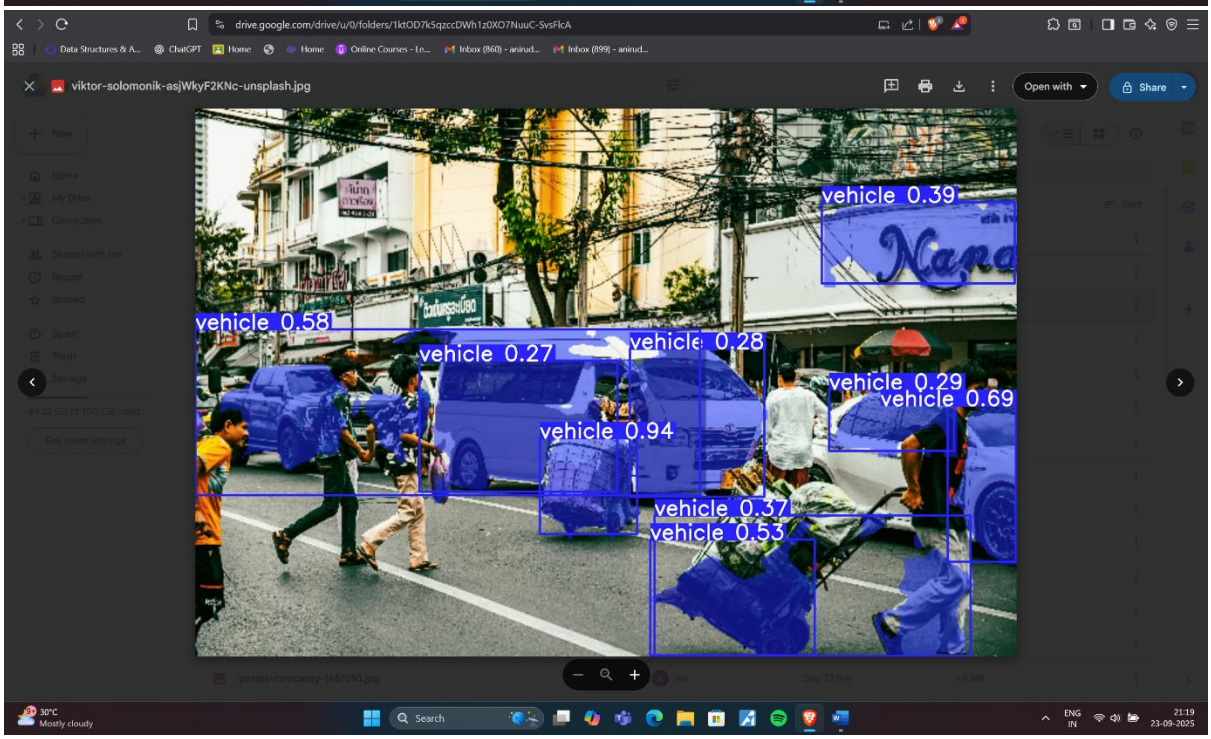




4. Evaluation

- Model evaluated on 40 test images + 26 validation images.
- Performance measured using **IoU, precision, recall, mAP**.
- Two demo videos were successfully processed through the model.



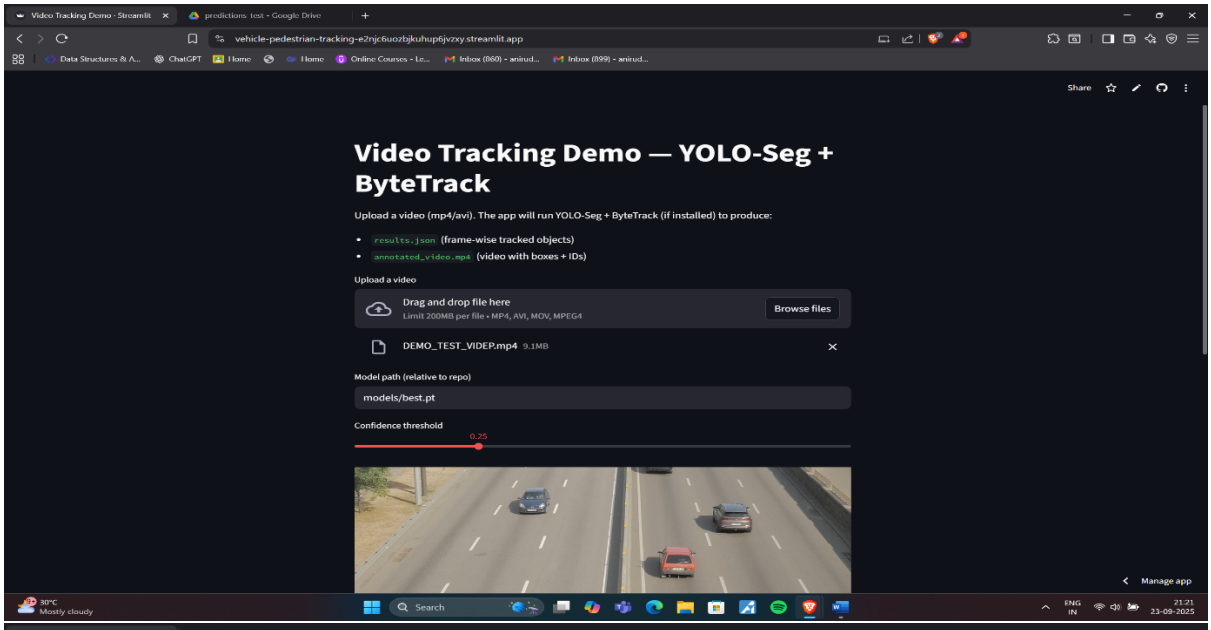


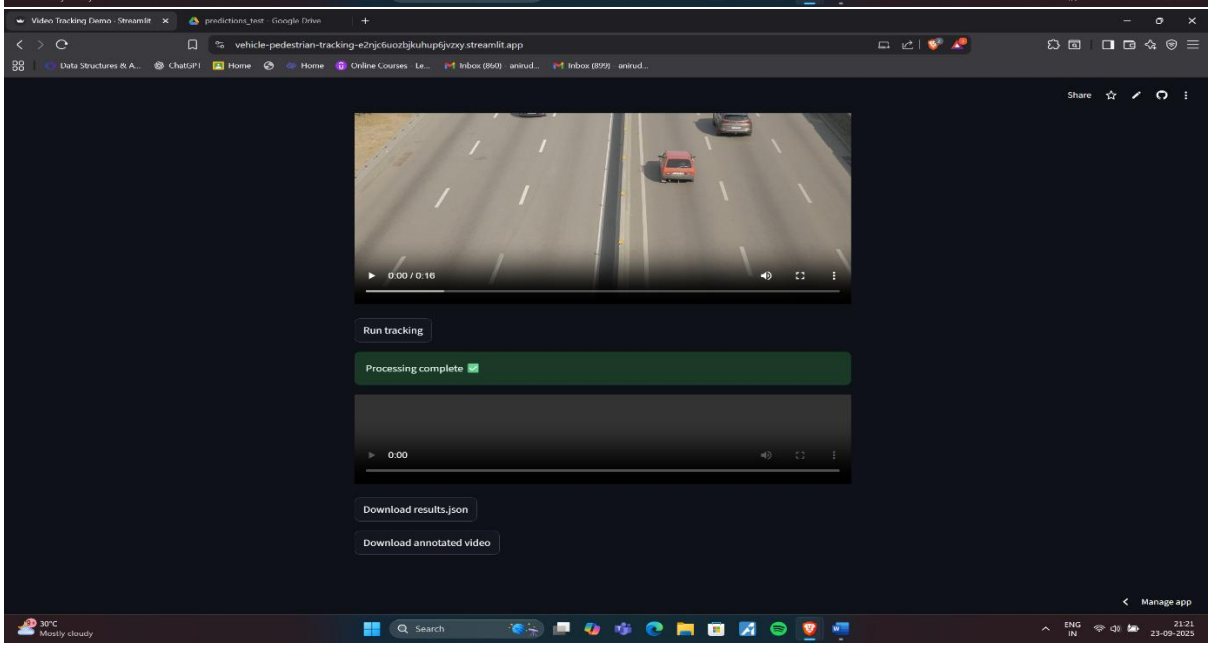
5. Web Application (Video Tracking Demo)

- **Framework:** Streamlit.
- **Features:**
  - Upload a video (mp4/avi/mov).
  - Run inference with YOLOv8 + ByteTrack.

- Download outputs:
    - results.json (tracked objects with IDs, classes, bounding boxes, frame numbers).
    - annotated_video.mp4 (tracking visualization).

- **Design:** Clean, minimal UI with download options.





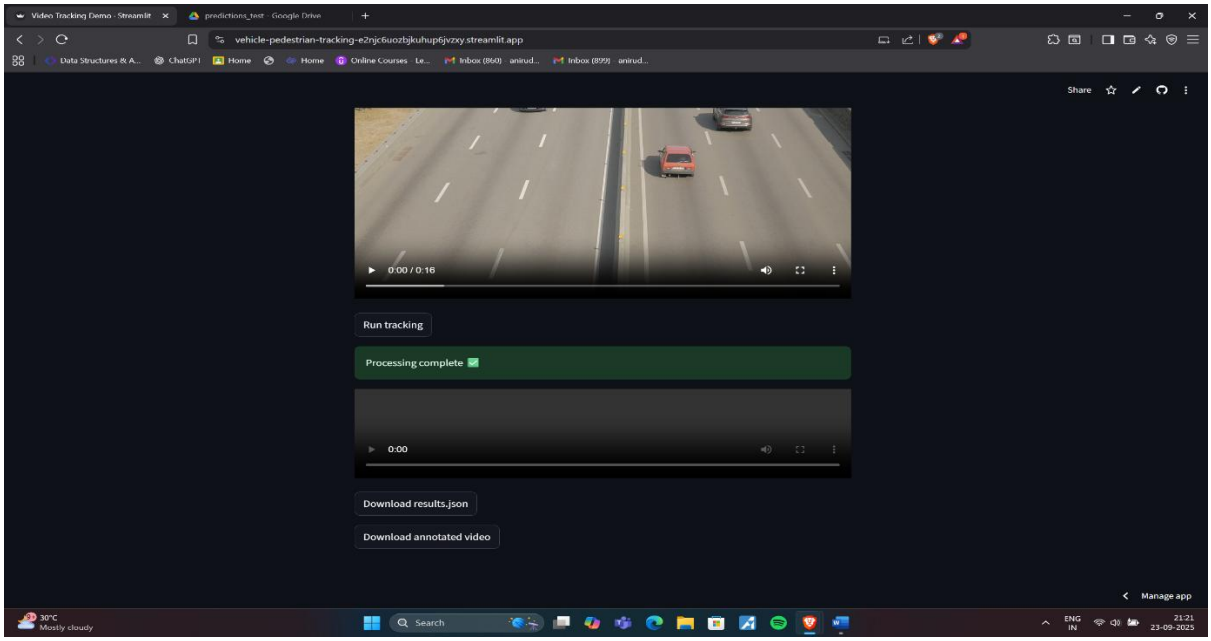
---

6. Deployment

- **Platform:** Streamlit Cloud.
- **Challenges faced:**
    - OpenCV (cv2) import errors due to Python 3.13.

- Fixed by pinning opencv-python-headless==4.10.0.84 and adding packages.txt with system dependencies (libgl1, libglib2.0-0).

- **Final Result:** Working live demo link available.
- **Live Demo Link:** https://vehicle-pedestrian-tracking-e2njc6uozbjkuhup6jvzxy.streamlit.app



---

7. Challenges & Resolutions

Challenge	Resolution
Model weights stored as .zip	Extracted and ensured correct best.pt available.
Earlier webapp attempt failed	Rebuilt from scratch with clean Streamlit setup.
cv2 import error on Streamlit Cloud	Used opencv-python-headless + system packages.
Session state bug in Streamlit	Fixed with st.session_state to keep download buttons visible.

---

8. Guide for Reproduction

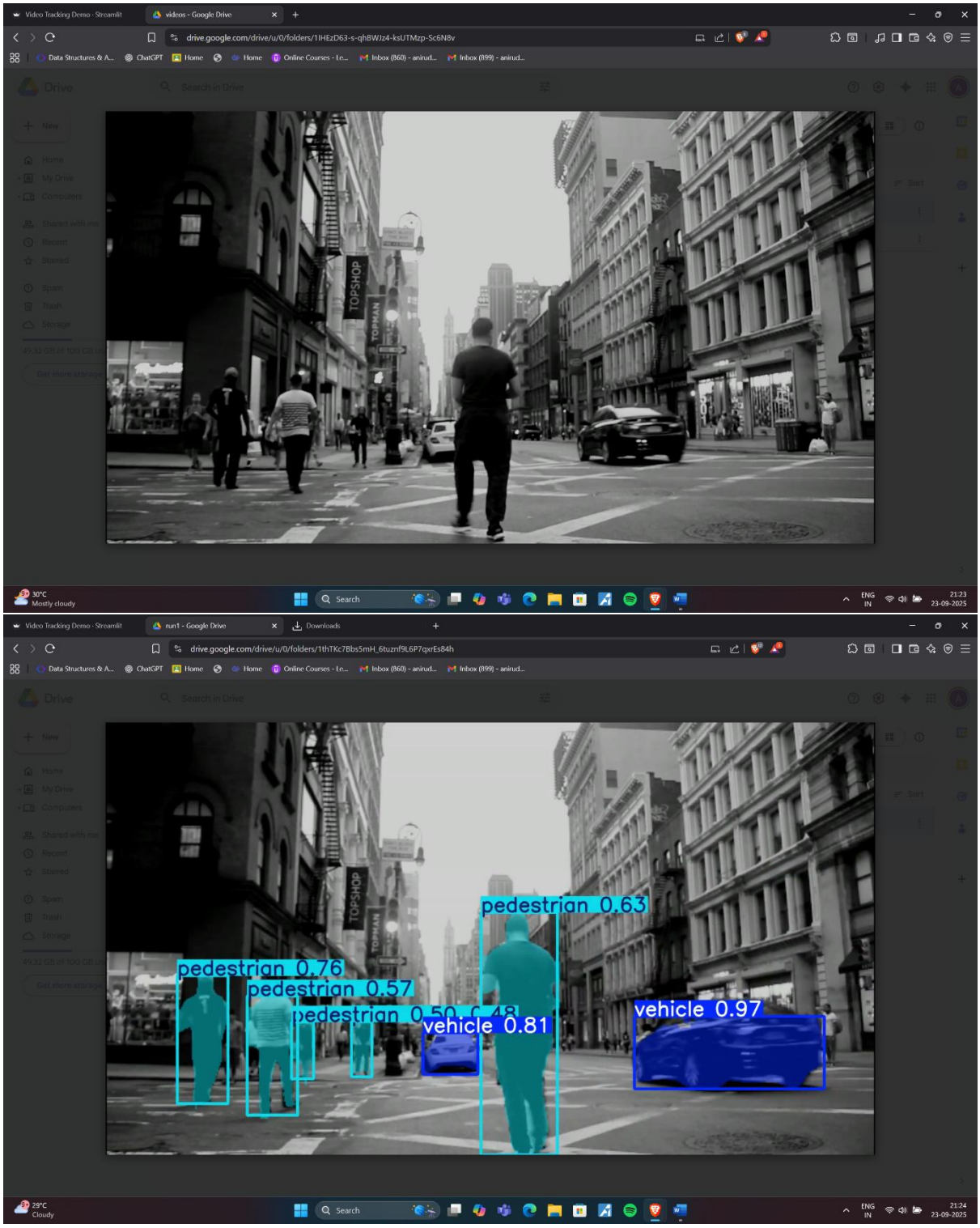
Steps for anyone to build their own tracker:

1. Collect and annotate dataset on Labellerr.
2. Export YOLO format dataset.
3. Train YOLOv8-seg on Colab or local GPU.
4. Save best weights as models/best.pt.
5. Clone repo and install dependencies (pip install -r requirements.txt).
6. Run streamlit run app.py.
7. Upload video → run inference → download results.
8. For cloud deployment:
    - Add requirements.txt and packages.txt.
    - Push repo to GitHub.
    - Deploy on Streamlit Cloud.

---

9. Results Summary

- Successfully built a trained YOLOv8 segmentation model.
- Integrated with ByteTrack for robust video tracking.
- Delivered a working Streamlit web application.
- Live demo available on Streamlit Cloud.



---

10. Conclusion

This project successfully achieved its objectives of detection, tracking, and deployment. It also highlighted the practical challenges of environment compatibility (Python versions, dependencies) and the importance of modular, clean code design.

The workflow can be easily extended to additional classes (e.g., bicycles, buses) and scaled for real-time streaming applications.

Appendix

- **GitHub Repo:** https://github.com/ANIRUDH-SHARMA-25/vehicle-pedestrian-tracking
- **Live Demo Link:** https://vehicle-pedestrian-tracking-e2njc6uozbjkuhup6jvzxy.streamlit.app

."