Project Journey & Final Summary

Journey
The project began with dataset preparation. Images and annotation labels were organized into a YOLO-compatible structure, with separate *train* and *val* folders and a data.yaml configuration file defining two classes: pedestrian and vehicle. Next, a YOLOv8 segmentation model (yolov8n-seg.pt) was fine-tuned for 50 epochs on this dataset. During the training, several issues arose such as missing 'val:' keys in the YAML, incorrect dataset splits, and mismatched class IDs. These were resolved by restructuring the dataset, correcting YAML format, and enforcing consistent label IDs. Evaluation of the trained model was carried out using Ultralytics validation utilities, producing metrics such as Precision, Recall, and mAP. These results confirmed that the model was learning effectively. The next stage integrated ByteTrack for multi-object tracking across frames. This ensured objects such as pedestrians and vehicles retained consistent IDs across video frames. Visualization was improved by adding bounding boxes, segmentation masks, and labels onto test videos. Deployment was achieved by building a Streamlit-based web app. Users could upload videos, run YOLO+ByteTrack tracking, visualize outputs, and download both processed videos and a results.json file containing frame-wise tracked object details. Ngrok tunneling was used to make the demo accessible online. Finally, predictions were converted to COCO JSON format and uploaded to Labellerr, demonstrating seamless integration for annotation visualization.

Problems Faced & Resolutions
1. **YAML Errors** – Fixed by adding missing 'train' and 'val' keys.
2. **Dataset Split Issues** – Corrected directory structure with /train and /val folders.
3. **Class ID Mismatches** – Standardized labels to 0: pedestrian, 1: vehicle.
4. **No Predictions** – Ensured segmentation mode was enabled and retrained with correct data.
5. **Deployment Challenges** – Used Streamlit with Ngrok tunneling for easy demo access.

Guide for Others
1. Prepare your dataset in YOLO format (images + labels with polygons for segmentation).
2. Create a proper data.yaml with 'train', 'val', 'nc', and 'names' fields.
3. Fine-tune a YOLOv8 segmentation model on your dataset.
4. Validate performance using Ultralytics tools and export metrics.
5. Integrate ByteTrack with YOLO outputs for multi-frame tracking.
6. Build a web demo (e.g., Streamlit) to allow video uploads and tracking.
7. Export predictions to COCO JSON format and upload them to Labellerr for annotation management.

Final Summary
This project demonstrated the full pipeline of creating a custom object tracker using YOLOv8 segmentation + ByteTrack with Labellerr integration. The key steps included dataset preparation, training, evaluation, tracking integration, visualization, and deployment. **Deliverables include:**
- Trained YOLO model weights (best.pt)
- Processed tracking videos
- Results.json with tracked objects
- Model evaluation PDF
- Project Journey & Guide PDF
- Streamlit demo web app

The outcome is a scalable, reproducible, and user-friendly tracker that enables anyone to develop and deploy their own video-based tracking system with the help of Labellerr.