Assignment Solution - AI Software Engineer Internship

1. Data Collection

We will collect 150–200 raw images containing vehicles and pedestrians. The data will be sourced from publicly available datasets such as Cityscapes, BDD100K, or custom web-scraped images (with proper licenses). A sources.md file will be maintained with dataset links and license notes. We will also mix in a small set of self-captured images for diversity.

2. Data Annotation with Labellerr

Using Labellerr platform, we will create two projects: - Train Project: Annotate ~100 images with polygon masks for classes: vehicle, pedestrian. - Test Project: Upload ~50 images as test data. Later, upload model predictions for review.

3. Exporting Annotations

Annotations will be exported from Labellerr in YOLO-compatible JSON/YOLO format. These labels will be used directly for YOLO-Seg training.

4. Model Training (YOLO-Seg)

- Framework: Ultralytics YOLOv8-seg - Backbone: yolov8n-seg (lightweight for colab GPUs). - Training: ~100 epochs with train/val split (80:20). - Metrics: IoU, mAP, precision, recall will be logged. - Reproducibility: Seed setting and proper dataset splits maintained. Bonus: Data augmentation (flipping, scaling, brightness) and hyperparameter tuning will improve results.

5. Debugging Process

At least two real-world issues will be debugged, such as: 1. Handling class imbalance in dataset by oversampling underrepresented classes. 2. Fixing YOLO data loader errors caused by label format inconsistencies.

6. Inference & Evaluation

The trained model will be evaluated on the test set. Deliverables: - Predictions saved and visualized on test images. - Metrics such as confusion matrix, PR curve, IoU per class.

7. Integration with ByteTrack

YOLO-Seg will be integrated with ByteTrack for vehicle/pedestrian tracking in videos. - Input: Uploaded video. - Output: JSON file containing IDs, class, bounding boxes, and frame numbers. - Example output: {"id": 1, "class": "vehicle", "bbox": [x,y,w,h], "frame": 12}.

8. Web App Demo

A minimal Flask/Streamlit app will be developed: - Upload video. - Run YOLO-Seg + ByteTrack on frames. - Display tracked objects with IDs in video overlay. - Allow export of results.json.

9. Documentation & GitHub Repo

- GitHub repository with source code, training notebooks, and README (steps, decisions, problems, results). - README will include installation guide, training process, evaluation results, and usage of Labellerr. - PR will be raised to Labellerr/campushiring repo.

10. Deliverables

- GitHub link with full implementation. - Live demo link of the web app. - PDF report including dataset details, training setup, evaluation metrics, debugging steps, and summary.