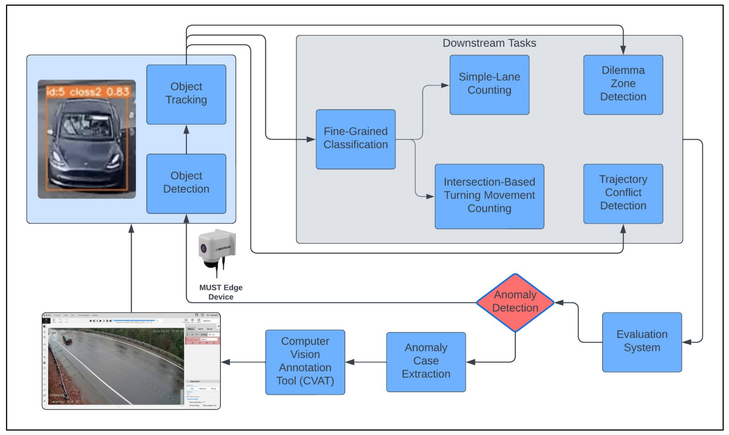
Achieved fine-grained classification for cars and trucks, with the final mAP on the evaluation dataset and test dataset being 0.454 and 0.854, respectively.

AutoML for Traffic Video Analysis System Using Query-based Learning

Objectives

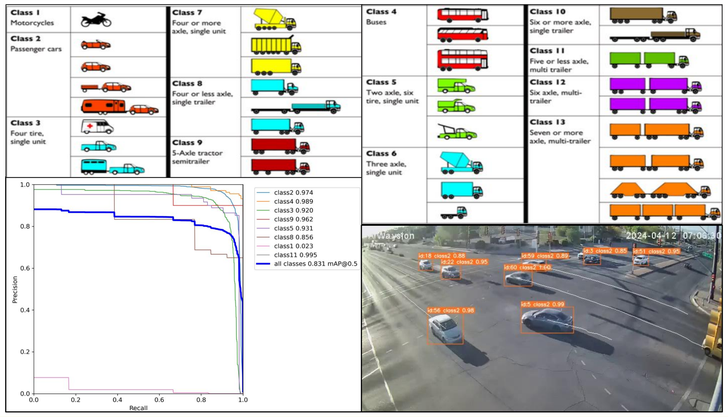
This project is dedicated to developing advanced computer vision AI models tailored for analyzing traffic videos, with a focus on vehicle counting and anomaly detection. Leveraging dynamic, state-of-the-art algorithms in **multiple object tracking**, our team aims to **enhance traffic monitoring systems**. Key objectives include: accurately counting vehicles along both simple-lane and complex intersection trajectories, enhancing detection of vehicles in dilemma zones, accurately identifying trajectory conflicts among vehicles, and implementing a sophisticated **query-based model** for detecting anomalies.

Project Pipeline



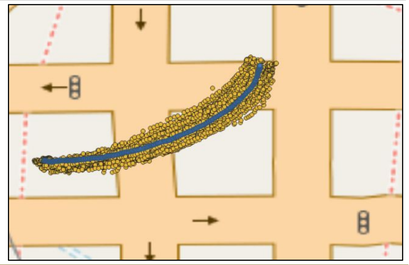
Object Detection & Tracking

- Single-Stage YOLOv7 was utilized for detecting 13 distinct vehicle classes, trained on the COCO dataset and 2444 custom annotated clips, achieving an mAP (mean average precision) over 0.8.
- Object tracking was conducted using StrongSORT and ByteTrack algorithms.



Intersection-Based Turning Movement Counting

- Used YOLOv7 to train an object detection model which has a mAP of 0.831.
- Combined detection and tracking result to implement counting algorithm based on typical trajectory annotation to assign each vehicle to different lanes. Below is the trajectory matching result for typical trajectory 0.
- Using camera calibration to achieve robust counting results based on 3D tracking.

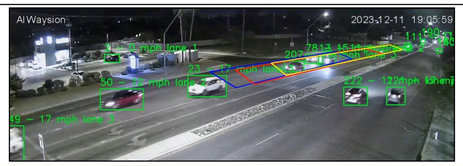


Simple Lane-Based Trajectory Counting / Fine-Grained Classification

- Achieved fine-grained classification based on 2444 custom annotated clips for cars and trucks, with the final mAP on the evaluation dataset and test dataset being 0.454 and 0.854.
- For vehicle counts, we compared five counting regions at different locations. By analyzing the average confidence of the counts at different locations, we obtained the best region for counting

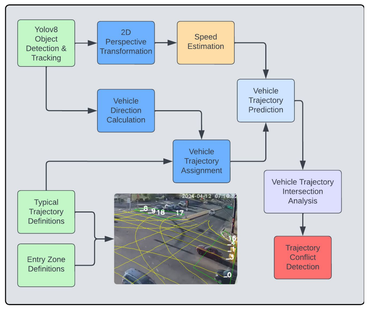
Dilemma Zone Detection Improvement

- Annotating the videos under low brightness conditions as dataset.
- Putting the annotated dataset into YOLOv7 model to train and having 0.89 mAP value.
- Using camera calibration to transfer 2D results to 3D.
- Implementing lane assignment and speed estimation algorithm to get the vehicles' lane and speed information.
- Capture the vehcles that unbale to stop safely before the traffic lights during yellow phase



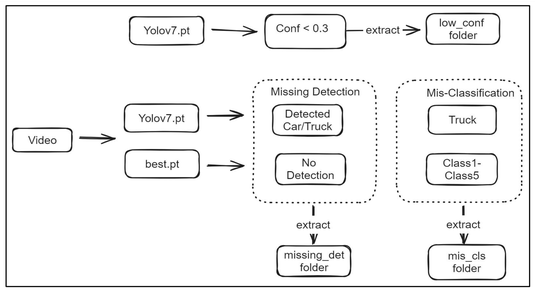
Trajectory Conflict Detection

This conflict detection model uses a custom-trained YOLOv8 model to track objects in video, focusing on detecting near-miss incidents between vehicles. It analyzes video frames to track vehicle positions and velocities, applying a perspective transformation to map 2D coordinates into a normalized space for precise analysis. The script identifies potential collisions by checking if vehicle trajectories intersect and calculates real-time speeds. It also detects when vehicles enter specific zones, predicts future positions based on typical movement patterns, and enhances collision detection by predicting path intersections and calculating the time to collision (TTC) using current velocities and trajectories.



Anomaly Detection & AutoML

- To enhance our model and minimize manual data collection and labeling, we introduced AutoML. Our system automatically detects anomalous frames, and assigns pseudo labels to them for subsequent model fine-tuning. We utilize two detectors for anomaly detection:
- **Yolov7**: Trained on the COCO dataset, it accurately detect vehicles as class 2 or trucks
- **best.pt**: Our custom model that performs fine-grained vehicle classification
- Comparing both detectors' results effectively identifies anomalous frames.



Future Work, References, and Acknowledgments

- Further refinement on anomaly detection.
- Accounting for pedestrians in trajectory conflict detection.
- Finalize script for generating typical trajectories based on tracking data.

References:
[1]Ultralytics, "Track," docs.ultralytics.com. https://docs.ultralytics.com/modes/track/
[2]bharath, "bharath5673/StrongSORT-YOLO," GitHub, May 07, 2024. https://github.com/bharath5673/StrongSORT-YOLO/tree/main (accessed May 15, 2024).

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: Dr. Hung-Min Hsu, Yiran Zhang, Dr. Wei Sun, Prof. Wei Cheng, Mingfei Chen

SPONSOR: AIWaysion