**Traffic rule prediction**

This project utilizes advanced YOLOv5 and YOLOv7 deep learning models for real-time traffic monitoring and safety assessment. It detects various types of vehicles, estimates vehicle counts and speed, and identifies motorbike helmet compliance. Leveraging video feeds, the system detects traffic flow patterns and highlights safety violations through "helmet" annotations for compliant riders. Its scalable design has applications in smart cities, traffic enforcement, and accident prevention. By comparing YOLOv5 and YOLOv7, the project identifies the optimal solution for diverse traffic scenarios while ensuring high-performance metrics like accuracy, precision, and recall to optimize overall system efficiency.

**Week 1 – 2:**

* Project Setup & Initial Planning - Define objectives and success criteria.
  + Install required libraries (cv2, YOLO, TensorFlow, etc.).
  + Set up Python 3.7.0 environment.
* - Download and validate the dataset from Kaggle. - Project plan
  + Initial setup verification

**Week 3 – 4:**

* Data Understanding and Preprocessing - Explore the Helmet Detection dataset using pandas and visualization tools.
  + Data cleaning, shuffling, and normalization.
* - Perform train-test split using sklearn.model\_selection. - Cleaned and pre-processed dataset
  + Data preprocessing report

**Week 5 – 6:**

* Data Augmentation & Annotation Review - Apply data augmentation (flipping, scaling, rotation) to enhance training samples.
  + Verify PASCAL VOC bounding box annotations.
* - Generate augmented dataset. - Augmented training dataset
  + Verified annotations

**Week 7 -8:**

* Model Training: YOLOv5 and YOLOv7 - Initialize YOLOv5 and YOLOv7 configurations.
  + Set hyperparameters: learning rate, batch size, input image size, and epochs.
* - Train both models on the dataset. - Trained YOLOv5 and YOLOv7 models and Training Logs

**Week 9:**

* Intermediate Evaluation and Debugging - Evaluate initial results using accuracy, precision, recall, and F1-score.
* - Identify and resolve any training issues (overfitting/underfitting). - Initial performance evaluation report

**Week 10:**

* Traffic Speed Estimation Integration - Integrate traffic speed estimation using frame difference and vehicle movement tracking.
* - Ensure YOLO detections are linked with speed calculations. - Speed estimation module integrated

**Week 11:**

* Helmet Compliance Detection Module - Develop submodule to check helmet compliance using bounding box verification.
* - Display real-time “helmet” annotations for compliant riders. - Helmet compliance detection module

**Week 12:**

* Comprehensive Model Evaluation - Compare YOLOv5 and YOLOv7 on key metrics (accuracy, precision, recall, F1-score).
* - Evaluate hardware efficiency (inference speed). - Comparison report of YOLOv5 vs YOLOv7

**Week 13:**

* Optimization and Fine-Tuning - Adjust hyperparameters based on evaluation feedback.
* - Optimize performance for real-time applications (confidence threshold, batch size, etc.). - Optimized model configurations

**Week 14:**

* System Integration and Testing - Integrate vehicle detection, speed estimation, and helmet compliance into a unified system.
* - Conduct end-to-end testing with different video inputs. - Fully functional integrated system

**Week 15:**

* Final Evaluation and Validation - Perform validation on unseen traffic videos.
* - Collect final metrics (accuracy, speed estimation accuracy, and violation detection). - Final performance report

**Week 16:**

* Project Report and Submission - Prepare technical documentation (methods, results, comparisons, and recommendations).
* - Submit project deliverables. - Final project report
  + Code Overview and documentation