Engineering Track: Real-Time Traffic Violation Detection and Plate Recognition

Authors: Mahi Mann(2022272), Himanshu Raj(2022216), Aarya Khandelwal(2022007)

Problem Statement and Scope:

Problem Statement:

Traffic violations such as running red lights, speeding, and illegal turns are a significant cause of accidents and congestion on the roads. These violations often go unnoticed and are difficult to enforce in real time, putting everyone on the road at risk. Manual monitoring of traffic violations is slow and inefficient, making an automated solution a requirement for road safety..

This project aims to develop a computer vision-based system which will consists of the following key components:

- **Traffic Violation Detection:** Automatically detecting violations such as running red lights, speeding, and illegal turns.
- **Number Plate Recognition:** Capturing and reading the vehicle's number plate to identify and track violators.

Project Scope:

- Detecting traffic violations in urban areas and highways, with a **focus on high-traffic zones** like neighborhoods and major roads.
- The system will address challenges like **blurry footage or obscured plates**, though performance may be impacted in such cases.
- The system will be tested to handle challenges such as **poor lighting, bad weather conditions** (e.g., rain, fog), and camera quality, which may affect detection accuracy.

Potential Users:

- **Traffic Authorities:** For monitoring and enforcing road safety rules.
- **Municipal Corporations:** For improving city traffic management and reducing accidents.
- **Insurance Companies:** To assess risk based on traffic rule violations.
- **Public Awareness Platforms:** Educating the general public about road safety.

Interface & Technology:

- A simple **web application** built using **Django** for the backend to manage and display real-time traffic violation data.
- The front-end can be developed using **HTML, CSS, and JavaScript** to show dashboards, traffic violation stats, and alerts.

Dataset and Evaluation Metrics:

Dataset

- **AI City Challenge 2023 Dataset**
  - Source: AI City Challenge 2023
  - Description: A large-scale dataset for traffic incident detection, tracking, and behavior analysis.

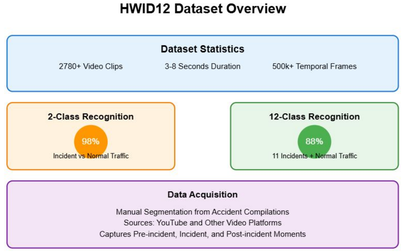
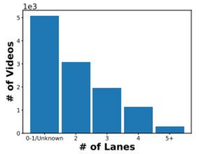
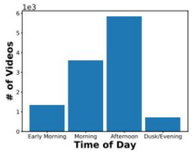


fig: dataset overview of HWID12

- **HWID12 Highway Incidents Detection Dataset**
  - Source: Kaggle
  - Description: A dataset containing annotated videos of highway incidents, including accidents, stopped vehicles, and congestion.
- **Traffic Surveillance Dataset (SciDB)**
  - Source: SciDB
  - Description: A comprehensive dataset of traffic surveillance footage for vehicle and anomaly detection



Percentage of videos wrt to number of lanes



Percentage of videos wrt time of the day

Screenshot of a video footage from HWID12 dataset.

Ref: Traffic Surveillance Dataset (SciDB)

Evaluation Metrics

- **Precision & Recall**: Measures the accuracy of incident detection.
- **F1 Score**: Harmonic mean of precision and recall, ensuring balanced performance evaluation.
- **Mean Average Precision (mAP)**: Used for object detection performance.
- **Intersection over Union (IoU)**: Evaluates the overlap between predicted and actual bounding boxes.
- **Inference Time**: Measures the speed of detection models.

Related Work:

- **Traffic Signal Violation Detection Using Computer Vision**
  - *(Ref: ResearchGate: Traffic Signal Violation Detection)*
  - This paper proposes a system for detecting red-light violations by leveraging object detection and tracking methods on **real time video footage.**
- **Self-Supervised Learning for Traffic Video Analysis**
  - *(Ref: arXiv:2209.12386)*
  - This study looks at a technique called **self-supervised learning** where the system learns from unlabeled data helping the system find patterns and make sense of the traffic videos on its own.
- **An Automatic Number Plate Recognition System**
  - (Ref: ResearchGate: 353596846)
  - Uses **ANPR algorithms like Haar Cascade Classifiers** or Deep Learning-based models (e.g., CNNs) for license plate recognition, which will be crucial for tracking and recognizing license plates of violaters in the system.

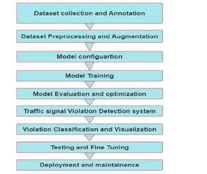


Fig.1. Control flow of the model for Traffic Signal Violation Detection System
Reference: (Ref: ResearchGate: Traffic Signal Violation Detection)

These studies provide insights into various methodologies for incident and violation detection, forming the foundation for our approach.

Requirements:

- **Development Environment**:
  - Python 3.10+
  - Jupyter Notebooks or any IDE (e.g.VSCode)
- **Hardware Requirements:**
  - For training model: Minimum 8 GB RAM, GPU (Optional, for faster training)
  - Storage: 10 GB free space for dataset and models
- **Possible Solutions:**
  - Dataset Storage: Use Google Colab (12 GB) or Kaggle (30 GB) for storing datasets.
  - Model Training: Leverage online GPUs or lab servers for training models
  - Web Deployment: Use Flask or Django for deploying the web application.

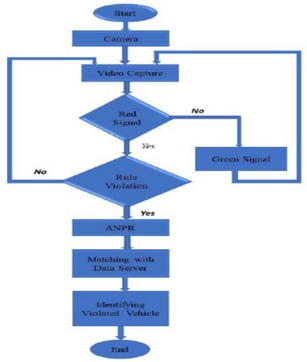


Fig. 2. System Flow Chart for the overall system

Reference: (Ref: ResearchGate: 353596846)