A Computer Vision-Based System for Real-Time Vehicle Speed Detection and Automated Complaint Logging

Meghana M[1], Mrs. Shruthi M T[2]

[1] Assistant Professor, Department of MCA, BIET, Davanagere

[2] Student, 4th Semester MCA, Department of MCA, BIET, Davanagere

Abstract: Vehicular speeding, particularly in residential and school zones, is a major public safety concern. Traditional speed enforcement methods are often resource-intensive and provide only sporadic coverage. This paper presents a low-cost, automated system that leverages computer vision to monitor traffic, detect speeding vehicles, and log complaints in real-time. The system processes a video feed from a standard camera to perform vehicle detection and tracking using the YOLO (You Only Look Once) and Deep SORT algorithms, respectively. By performing a perspective transformation via camera calibration, the system accurately maps pixel coordinates to real-world distances, enabling robust speed estimation. When a vehicle is detected exceeding a user-defined speed limit, the system automatically logs a "complaint" or incident report, which includes the timestamp, estimated speed, and video evidence, into a database. This data is made accessible through a web-based interface. This framework offers an accessible and data-driven tool for communities and local authorities to monitor traffic safety, identify speeding hotspots, and justify the need for traffic calming measures.

Keywords: Computer Vision, Speed Detection, Object Tracking, YOLO, Deep SORT, Intelligent Transportation Systems (ITS), Complaint Logging, Traffic Safety.

I. INTRODUCTION

Traffic safety is a critical component of urban planning and management. Speeding is a leading contributor to the severity of traffic accidents, especially in vulnerable areas like residential neighbourhoods, school zones, and hospital perimeters. While law enforcement agencies employ radar and LiDAR guns for speed enforcement, these methods require a significant allocation of personnel and can only cover a limited area at any given time. This leaves many local roads unmonitored, allowing unsafe driving behaviors to persist.

There is a growing need for cost-effective, continuous, and scalable solutions for traffic monitoring. The proliferation of low-cost digital cameras and the remarkable advancements in computer vision provide a powerful opportunity to develop such systems. By analysing video streams, an intelligent system can not only detect the presence of vehicles but also measure their dynamic properties, such as speed and trajectory.

This paper proposes a novel, end-to-end system that goes beyond simple speed detection. We present a framework that not only identifies speeding vehicles in real-time but also automatically generates a complaint or incident log. This log, complete with evidentiary snapshots or clips, serves as a quantitative record of unsafe driving behaviour. Such data can be invaluable for residents' associations appealing for traffic calming measures (like speed bumps) or for municipal authorities planning targeted enforcement campaigns.

The primary contributions of this work are:

1. The design of an integrated system that combines real-time vehicle speed detection with an automated complaint logging mechanism.
2. The application of state-of-the-art deep learning models (YOLO and DeepSORT) for robust vehicle detection and tracking in real-world traffic scenes.
3. The development of a practical workflow for camera calibration and speed estimation that can be deployed using standard, non-specialized camera hardware.

4.The conceptualization of a web-based backend for easy access and review of logged incidents.

II. RELATED WORK

The development of Intelligent Transportation Systems (ITS) has been a key area of research, with vehicle speed detection being a fundamental task.

Traditional, non-vision-based methods for speed detection include inductive-loop detectors embedded in the pavement and radar/LiDAR systems [1]. While highly accurate, these solutions involve significant installation costs, infrastructure modification, and maintenance, making them unsuitable for widespread, community-led deployment.

Computer vision-based approaches offer a more flexible and cost-effective alternative. Early vision systems relied on background subtraction or frame differencing to detect moving objects [2]. These methods, however, were highly susceptible to environmental noise such as changing illumination, shadows, and camera movement.

The advent of machine learning brought more robust feature-based methods, using descriptors like Histogram of Oriented Gradients (HOG) combined with classifiers like SVMs. The true revolution, however, came with deep learning. Convolutional Neural Networks (CNNs) have set new benchmarks for object detection. Single-stage detectors like YOLO (You Only Look Once) [3] and SSD (Single Shot Detector) provide an excellent balance of speed and accuracy, making them ideal for real-time video processing.

For speed detection, identifying a vehicle is not enough; it must be tracked across consecutive frames. The SORT (Simple Online and Realtime Tracking) algorithm [4] introduced an efficient framework using a Kalman filter for motion prediction and the Hungarian algorithm for data association. Its successor, DeepSORT [5], significantly improved tracking robustness by adding a deep, appearance-based metric. This helps maintain a vehicle's identity even after temporary occlusions, which are common in traffic. Several studies have successfully used these techniques for speed estimation [6].

Our work differentiates itself by integrating these advanced vision techniques into a complete, application-focused system. While most academic research stops at demonstrating speed detection accuracy, our framework extends to the practical problem of logging and reporting these violations, creating a tool with direct societal utility.

III. METHODOLOGY

The proposed system is architected as a modular pipeline that processes a video stream to generate a structured log of speeding incidents.

A. System Architecture

The system operates on a continuous loop, processing each frame from a video feed. The workflow is as follows:

1.Video Input: A frame is captured from a static camera monitoring a road.

2.Vehicle Detection: A pre-trained YOLO model is used to detect all vehicles in the frame, drawing a bounding box around each one.

3.Vehicle Tracking: The DeepSORT algorithm takes the detections from YOLO, assigns a unique ID to each vehicle, and tracks its position across subsequent frames.

4.Speed Estimation: Using a pre-calculated perspective transformation matrix, the vehicle's position in the image is mapped to real-world coordinates. Its speed is calculated based on the distance traveled over a known number of frames.

5.Violation Check & Complaint Logging: If the estimated speed exceeds a user-defined threshold, the system triggers the complaint logging module.

6.Data Persistence and Access: The logged incident is stored in a database and can be reviewed via a web interface.

B. Camera Calibration and Perspective Transformation

To measure speed accurately, we must convert measurements from the 2D image plane (pixels) to the 3D real world (meters). This is achieved through camera calibration.

A one-time setup process is performed where four or more points on the road surface with known real-world coordinates are selected in the camera's view. For example, the corners of a marked-out rectangle of a known size (e.g., 5 meters by 2 meters).

These source points (in pixels) and their corresponding destination points (in real-world units) are used to compute a perspective transformation matrix.

During operation, this matrix is used to transform the centroid of a vehicle's bounding box from its pixel location to its real-world position on the road plane.

C. Speed Estimation Module

With the ability to track an object and know its real-world position in each frame, speed can be calculated.

1.The system tracks the real-world distance a vehicle (identified by its unique ID) travels between two consecutive frames.
2.The time elapsed between these frames is known from the video's frame rate (e.g., at 30 FPS, the time between frames is 1/30 seconds).
3.The speed is calculated by dividing the distance traveled by the time elapsed. To improve stability, the speed can be averaged over a small window of frames.

D. Automated Complaint Logging System

This module forms the core application layer of the system.

Trigger Condition: A user sets a speed limit for the monitored road (e.g., 40 km/h). The system continuously compares the estimated speed of each tracked vehicle against this limit.

Logging Action: When a violation is detected, the system creates an incident record. This record is a structured data entry containing:

A unique incident ID.
The date and timestamp of the violation.
The estimated speed of the vehicle.
The predefined speed limit for that road.
A reference to the evidentiary material (e.g., a file path).

Evidence Capture: Simultaneously, the system saves a short video clip or a series of still images of the offending vehicle as it passes through the detection zone. This evidence is linked to the incident record.

Database and Web Interface: The incident records are stored in a database (e.g., SQLite for simplicity or PostgreSQL for scalability). A web application, built using a framework like Flask or Django, provides a user interface to query, view, and manage the logged complaints.

IV. RESULTS AND DISCUSSION

This section describes the expected functional outputs of the system, illustrated through descriptions of the user-facing visual components.

A. Real-Time Monitoring Interface

The primary output of the core vision system is an annotated video stream.

A snapshot would show a live video feed of a road. Each detected vehicle would have a bounding box drawn around it. The box would be labeled with the vehicle's unique tracking ID and its real-time estimated speed. Vehicles exceeding the speed limit could be highlighted with a different color (e.g., red) for immediate visual feedback.

B. Web-Based Complaint Log Dashboard

The web interface provides a user-friendly portal to the collected data.

Dashboard View: A snapshot of the main dashboard would display summary statistics, such as "Total Incidents Today," "Highest Speed Recorded," and a chart showing the distribution of incidents by time of day.

Incident Log Table: Another snapshot would show a detailed, sortable table of all logged complaints. Columns would include Date, Time, Detected Speed, Speed Limit, and a "View Evidence" button. Clicking this button would open the corresponding image or video clip for review.

C. Discussion

The proposed system presents a practical and powerful tool for community-driven traffic safety monitoring. By automating the entire process from detection to reporting, it provides objective, actionable data.

However, the system is subject to several challenges and limitations:

Calibration Accuracy: The accuracy of the speed estimation is fundamentally dependent on the precision of the initial camera calibration. Any error in measuring the real-world reference points will propagate through the calculations.

Environmental Conditions: Performance can be degraded by poor lighting (especially at night), adverse weather (heavy rain, fog), or shadows, which can affect the accuracy of the object detector.

Occlusion: In heavy traffic, one vehicle may temporarily block the camera's view of another, potentially causing the tracker to lose its target and resulting in a missed detection.

Legal Standing: It is crucial to note that this system, as designed, is an *informational and data-gathering tool*, not a system for legal enforcement. Without official certification and robust license plate recognition, its "complaints" do not constitute legal violations but rather serve as data points to justify further action.

V. CONCLUSION AND FUTURE WORK

This paper has detailed the framework for a computer vision-based system that not only detects vehicular speed but also automates the process of logging complaints for violations. By integrating state-of-the-art deep learning models into a practical, application-focused pipeline, the system offers a low-cost, scalable solution for improving traffic safety. The automated generation of an evidence-backed incident log empowers local communities and authorities with the data needed to make informed decisions about traffic management. Future enhancements could further increase the system's utility and robustness:

1.Automated License Plate Recognition (ALPR): Integrating an ALPR module to automatically read and record the license plates of violating vehicles (subject to local privacy laws and regulations).

2.Advanced Traffic Analytics: Expanding the system to detect other dangerous behaviors, such as illegal turns, running stop signs, or vehicle classification for traffic flow analysis.

3.Real-Time Alerting: Implementing a notification system to send real-time alerts (e.g., via email or SMS) to a designated user or authority when a severe speeding event occurs.

4.Edge Deployment: Optimizing the models to run on low-power edge computing devices (e.g., NVIDIA Jetson Nano, Raspberry Pi with an accelerator) for a fully standalone, field-deployable unit.

9

REFERENCES

[1] L. A. Klein, M. R. Kelley, and M. K. Mills, "Traffic Detection and Surveillance for Intelligent Transportation Systems," in *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 3, pp. 24-34, 2009.

[2] R. T. Collins et al., "A system for video surveillance and monitoring," in *Technical Report CMU-RI-TR-00-12*, Carnegie Mellon University, 2000.

[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016.

[5] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2017.

[6] V. A. S. and M. D. S., "Real-Time Vehicle Speed Estimation from a Moving Camera," in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.