Smart Traffic Violation Detection and Challan Issuance

Dr. S. Senthamizh Selvi*, Yuvaraj K†, S Varsha†, Varun Kumar R†
*Associate Professor, Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering, Chennai, India
†Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering, Chennai, India

I. Abstract

A. Purpose

This project develops a real-time, AI-based system for detecting traffic violations such as helmet non-compliance and stop-line crossing. It automates evidence collection, challan generation, and notification processes. The goal is to enhance traffic law enforcement efficiency, reduce manual intervention, and support safer urban mobility through intelligent, integrated technology.

B. Methodology

The system uses a YOLO-based object detection model for identifying violations and an OCR-based ANPR module to extract license plate numbers. A Flask backend manages violation data, while Stripe handles digital payments. Firebase tracks officer locations and sends FCM alerts. An Android app and Google Maps heatmaps support field enforcement.

C. Findings

Testing showed over 91% accuracy in both helmet and stop-line violation detection. The system demonstrated high precision, reliable alert delivery, and successful online challan payment processing. Real-time coordination between detection, alerting, and payment modules validated the system's effectiveness and scalability for smart city traffic enforcement.

D. Practical implications

The system reduces dependency on manual monitoring, minimizes human error, and speeds up fine processing. It enables quick response by notifying nearby officers and offers a seamless digital payment experience for violators. Designed for real-world use, it fits into existing urban traffic systems and supports smart governance initiatives.

E. Social implications

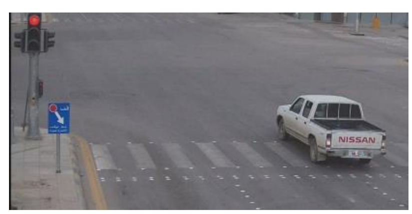
By promoting consistent enforcement and digital accountability, the system encourages safer driving habits. It reduces traffic accidents, increases public trust in law enforcement, and supports broader adoption of smart city technologies. Automated alerts and fair, transparent processes help create a culture of responsible urban mobility.

***Index Terms*—Traffic Monitoring, YOLO, Helmet Violation, Stop-Line Detection, Firebase, Stripe, Android App, Smart City**

II. Introduction

Urbanization and the rapid rise of motor vehicles have dramatically increased the complexity of traffic management systems in modern cities. Road safety, a critical component of urban governance, is often compromised due to traffic violations such as helmet non-compliance and stop-line jumping as depicted in Fig. 1. These infractions not only endanger the lives of violators but also pose risks to fellow commuters and disrupt traffic flow. As traffic volumes grow and law enforcement resources remain limited, the ability to monitor and enforce traffic regulations in real-time becomes increasingly difficult. Manual enforcement strategies are insufficient in densely populated urban settings, where thousands of vehicles pass through intersections every hour.

(a) Helmet Non-compliance



(b) Stop-line Crossing

Fig. 1: Common Traffic Violations

A major challenge in traffic law enforcement is the dependency on physical presence and human observation. Officers are expected to monitor multiple intersections, identify violations, document offenses, and initiate penalty processes—all in real time. This approach is not only labor-intensive but also prone to errors and inefficiencies. In many cases, violations go undetected, or data collected on-site is not processed effectively due to delayed manual handling. Furthermore, issuing challans (penalty notices) and ensuring fine payment adds to the administrative burden on enforcement agencies. These limitations highlight the need for automation in both violation detection and the post-detection workflow.

The evolution of technology, particularly in computer vision, cloud computing, and mobile communications, presents a unique opportunity to address these challenges as seen in Fig. 2. The integration of AI-driven surveillance systems, real-time notifications, digital payment gateways, and mobile apps can significantly enhance the efficiency and transparency of traffic law enforcement. Object detection models such as YOLO (You Only Look Once) have shown high accuracy in real-time image classification tasks, making them suitable for identifying helmet violations and stop-line breaches from traffic footage. When combined with OCR-based number plate recognition and cloud-based alert systems, these technologies can automate a large portion of the enforcement process.

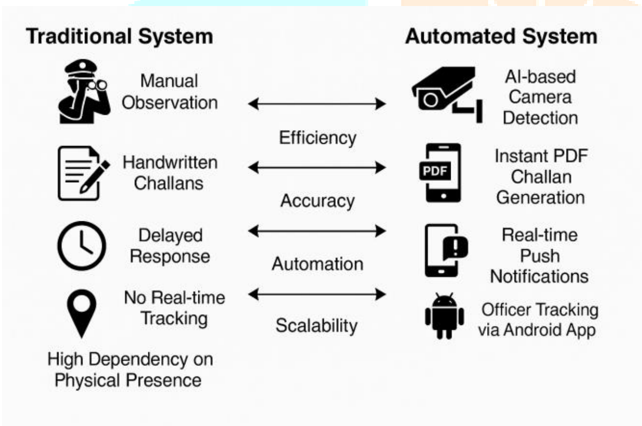


Fig. 2: Comparison of Traditional vs Automated Traffic Enforcement Systems

Cloud platforms like Firebase enable real-time communication and data sharing across devices, allowing location-aware systems to dynamically route information to nearby officers. Payment gateways such as Stripe facilitate secure and instant fine collection, reducing manual paperwork and encouraging digital compliance. On the field, Android applications can be leveraged by traffic personnel to receive violation alerts, verify offenses, and respond promptly—all without the need for physical observation at the time of the incident. These technologies together create a closed-loop system for real-time, automated traffic violation monitoring and enforcement.

As cities continue to adopt smart infrastructure, the need for intelligent traffic governance becomes paramount. Automated traffic monitoring not only improves compliance but also helps generate valuable data insights for urban planning and traffic flow optimization. By visualizing violation patterns through tools like heatmaps on platforms such as Google Maps, traffic authorities can identify high-risk zones and allocate resources accordingly. Such systems also contribute to public awareness and behavior correction through timely alerts and digital notifications. Ultimately, technology-enabled traffic enforcement can create safer roads, reduce the burden on law enforcement, and align with the larger vision of smart, connected cities.

III. Literature Review

S. Shobana and M. Jaya (2021) proposed an automated traffic violation monitoring system for detecting helmetless riders using deep learning. The authors utilized YOLOv3 for helmet detection and an OCR-based method for automatic number plate recognition (ANPR). The detected violations were then used to generate e-challans sent via email. Their work demonstrated improved accuracy but lacked real-time alerting and mobile integration for field officers.

R. Vishnupriya and M. Indumathi (2020) developed an AI-based surveillance system focused on license plate recognition and rider detection. They employed object detection for vehicle and helmet detection, followed by character segmentation for plate extraction. While effective in violation detection, their system did not support push notifications or geographic tagging of violations.

P. Sowmya and T. Venkat (2018) introduced a system using image processing techniques to monitor helmet use among two-wheeler riders. Their methodology involved color-based segmentation and edge detection. However, their work was limited by lighting conditions and failed to incorporate robust deep learning models or scalable infrastructure.

K. Divya and B. Jansi Rani (2019) explored traffic rule enforcement using a CNN-based detection system combined with OpenCV for video analysis. They successfully detected helmet violations and used OCR for number plate recognition. Yet, the study did not offer an integrated notification or payment system, making post-violation processing manual.

R. Akanksha and S. Anil Kumar (2019) presented an approach for monitoring violations using a Raspberry Pi-based system. Their lightweight solution performed basic image processing tasks and number plate recognition. Despite its low cost, the model lacked real-time processing capability and cloud-based alert mechanisms.

S. Sumathi and K. Jeyanthi (2021) proposed an enhanced traffic management system utilizing YOLO for detecting vehicles and pedestrians. Their system focused more on traffic flow than individual violations but showcased the benefits of using deep learning for visual traffic data analysis.

K. R. Harini and D. Kumar (2020) implemented an AI-driven solution that detects helmet usage and captures vehicle registration numbers from video frames. While the system reported good detection rates, it did not extend to user notification or integration with Android or cloud services.

M. Pooja and B. Kamalraj (2021) investigated the use of machine learning algorithms to predict and visualize high-violation areas. Their work incorporated spatial mapping but lacked a violation-specific detection pipeline or automated enforcement workflows.

IV. System Architecture

The proposed system is designed as a modular, cloud-connected, AI-driven traffic violation monitoring platform. It is composed of three major components: the desktop-based detection module, a centralized backend server, and a mobile application for field-level enforcement. Each module interacts through defined data flows, supported by cloud services for real-time communication, location tracking, and digital reporting [Refer Fig. 3].

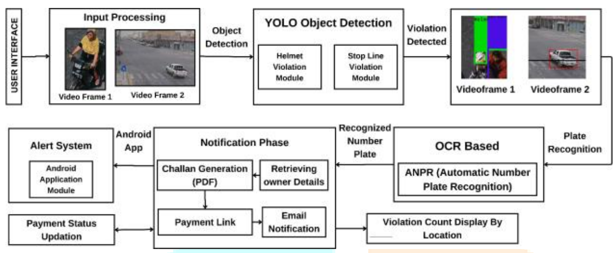


Fig. 3: Overall System Architecture

A. Detection and Recognition Module

The detection module is responsible for identifying two types of traffic violations—helmet non-compliance and stop-line crossing—from video input. Helmet detection is performed using a YOLO object detection model trained on two-wheeler rider datasets. Stop-line crossing is detected through grayscale image processing combined with spatial evaluation of the vehicle's position relative to a virtual stop line.

Once a violation is detected, the module uses an OCR-based Automatic Number Plate Recognition (ANPR) system to extract the vehicle's license plate number. This extracted data is used to query a local database to retrieve violator details, such as name, phone number, and email as seen in Fig. 4.

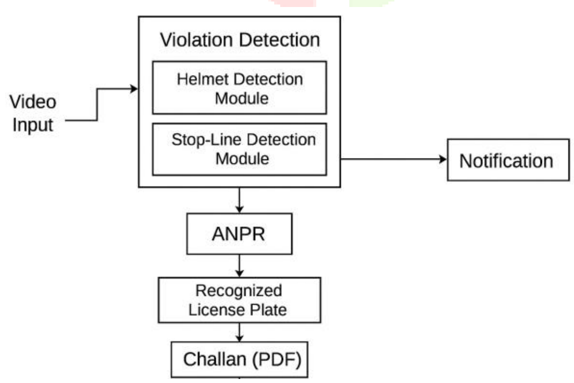


Fig. 4: Detection and Recognition Workflow

B. Backend Server and Communication Module

The backend system, developed using Flask, serves as the central coordinator for data handling and communication. It provides REST endpoints to initiate detection, generate and email PDF challans, and update violation logs. The server also integrates with external services such as the Stripe API for online payment processing and WhatsApp messaging for instant violator notification.

Firebase Realtime Database is used for tracking the GPS location of traffic officers. The server monitors this data and calculates proximity of officers to the violation using the Haversine formula. Officers within a specified radius (typically 250 meters) receive alerts via Firebase Cloud Messaging (FCM), including vehicle number, image, and location.

V. Proposed Work

This project proposes an integrated, real-time, automated traffic violation monitoring and enforcement system that combines artificial intelligence, mobile computing, cloud infrastructure, and modern notification mechanisms. The objective is to eliminate the limitations of traditional traffic enforcement methods by enabling efficient detection, accurate recognition, instant alerts, and secure digital fine collection—all coordinated across multiple platforms.

The system is composed of four primary functional modules: (i) helmet violation monitoring, (ii) stop-line crossing detection, (iii) mobile notification and enforcement via Android app, and (iv) online challan generation and payment processing. Each module is designed to function autonomously while contributing to the unified system architecture.

A. Helmet Violation Monitoring Module

The helmet violation module processes live or recorded video feeds using the YOLO object detection algorithm. It identifies two-wheeler riders and classifies them based on helmet usage. When a violation is detected, the frame is extracted and passed to an OCR-based Automatic Number Plate Recognition (ANPR) system to identify the violator's vehicle registration number.

Once identified, the data is stored and used to generate a challan in PDF format. The violation image, number plate, and details such as date and time are embedded into the report as seen in Fig, 5.

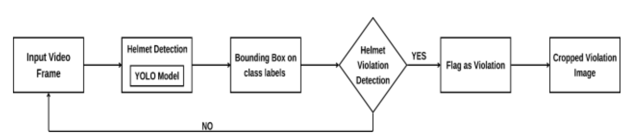


Fig. 5: Workflow of Helmet Violation Monitoring Module

B. Stop-Line Crossing Detection Module

This module uses grayscale image processing and line-based thresholding to detect vehicles that cross the stop-line during a red signal [Refer Fig. 6]. Once the vehicle crosses a predefined virtual line, a violation is flagged. The corresponding frame is captured and sent to the ANPR module for plate recognition.

The captured data is similarly used to generate a violation challan with image and license number details. The system ensures time-stamping and evidence tagging to support real-time enforcement.

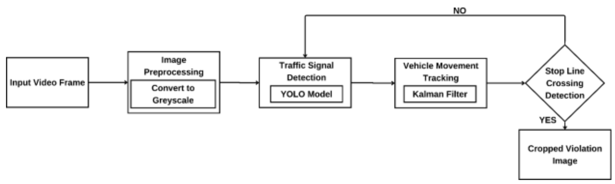


Fig. 6: Workflow of Stop-Line Crossing Detection Module

C. Notification and Alerting Module

The notification module ensures timely delivery of alerts to both violators and nearby traffic officers. Firebase Realtime Database is used to track the real-time GPS location of field officers through the companion Android app. Once a violation is detected, the system calculates officer proximity and sends alerts using Firebase Cloud Messaging (FCM) [Refer Fig. 7]. Violators receive alert messages through email and WhatsApp using the pywhatkit Python library. Additionally, Google Text-to-Speech (TTS) is used at the point of detection to provide audible warnings when a violation occurs in a live environment.

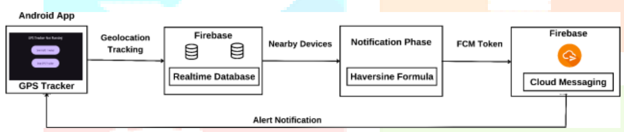


Fig. 7: Notification Flow

D. Online Payment and Challan Integration Module

The system integrates with the Stripe API to generate secure, unique payment links embedded within the challan PDF [Refer Fig. 8]. Once the violator receives the challan through email or WhatsApp, they can click the link to pay the fine online. Stripe's backend webhook is used to verify successful payment and update the violation status in the database.

This integration reduces manual intervention, encourages quicker compliance, and ensures a transparent and auditable enforcement process.

E. Backend Orchestration and Heatmap Visualization

All modules are coordinated through a Flask-based backend server, which manages routing, file generation, Firebase access, and notification dispatch. The system also visualizes violation data using Google Maps API. Real-time JSON feeds update heatmaps that display location-wise violation density, enabling authorities to identify hotspots and allocate resources accordingly as seen in Fig. 9.

Fig. 8: Online Challan Payment Workflow using Stripe API

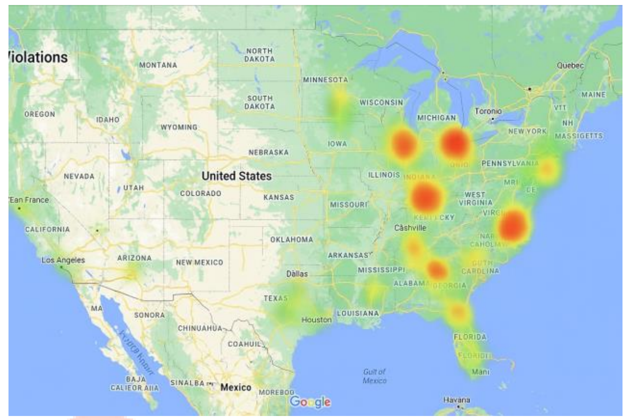


Fig. 9: Google Maps Heatmap of Violation Zones

F. Grayscale Conversion for Stop-Line Detection

To improve computational efficiency and detection accuracy, the system employs grayscale conversion as a preprocessing step in stop-line violation detection. Each frame from the traffic camera feed is converted from RGB to grayscale, significantly reducing data complexity while preserving essential information related to vehicle shape and contrast [Refer Fig. 10].

This conversion simplifies the pixel-wise comparison between consecutive frames and enhances edge-based detection along the virtual stop-line. A predefined threshold is applied to grayscale pixel intensities to detect motion across the stop-line during red signals. This method is particularly effective under variable lighting conditions, such as during nighttime or in areas with partial shadowing.

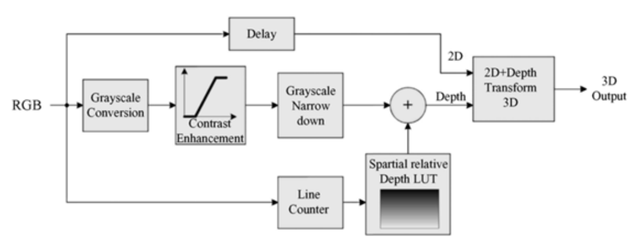


Fig. 10: Grayscale Conversion Workflow for Stop-Line Violation Detection

G. FCM-Based Alert Dispatch Mechanism

The system uses Firebase Cloud Messaging (FCM) to push real-time alerts to field officers when a traffic violation is detected. Each officer's live GPS coordinates are tracked via Firebase Realtime Database, and the system calculates their proximity to the violation location using the Haversine distance formula. Fig. 11 shows the workflow of the FCM based violation.

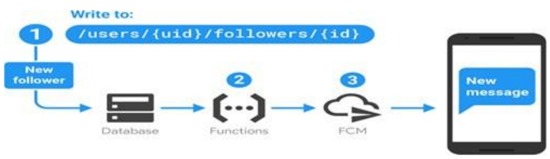


Fig. 11: Violation Alert Via FCM

If an officer is found within a defined radius (e.g., 250 meters), a detailed notification is immediately sent through FCM. The alert contains critical information such as the vehicle number, type of violation, timestamp, and a link to the image stored in the cloud. This ensures that the closest officer is informed promptly and can take necessary enforcement actions.

The use of FCM improves notification delivery speed and ensures reliable communication between the backend and Android application. It also supports background message handling, allowing the app to receive alerts even when not actively in use.

VI. Evaluation Metrics

To validate the performance of the detection modules, the system was evaluated using confusion matrices and key metrics such as accuracy, precision, recall, and F1-score. The evaluation was conducted on test datasets that simulated real-world traffic scenarios under varied lighting and movement conditions.

A. Helmet Violation Detection Evaluation

The helmet violation module was tested on a dataset of two-wheeler riders captured from traffic cameras. The system correctly identified 32 riders without helmets and 9 with helmets. However, it misclassified 4 actual violations as non-violations (false negatives), and 1 compliant rider was incorrectly flagged (false positive). Fig. 12 represents the confusion matrix of the Helmet Violation Detection where various different scenario based datasets are used for this prediction.

From the confusion matrix, the calculated performance metrics were:

- **Accuracy:** 91.11%
- **Precision:** 96.97%
- **Recall:** 88.89%
- **F1-score:** 92.73%

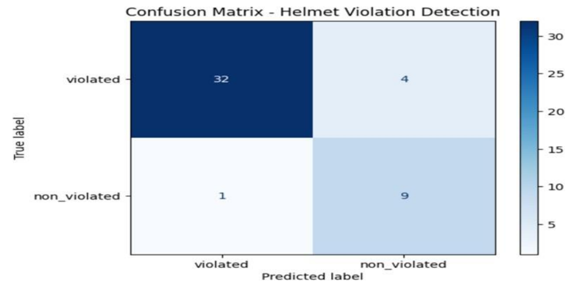


Fig. 12: Confusion Matrix for Helmet Violation Detection

B. Stop-Line Crossing Detection Evaluation

For the stop-line crossing module, 7 actual violations and 28 non-violations were considered. The system successfully detected 6 violations and 29 non-violations, with 1 false negative and 1 false positive. Fig. 13 represents the confusion matrix of the stop-line crossing detection where various different scenarios like night light and day light.

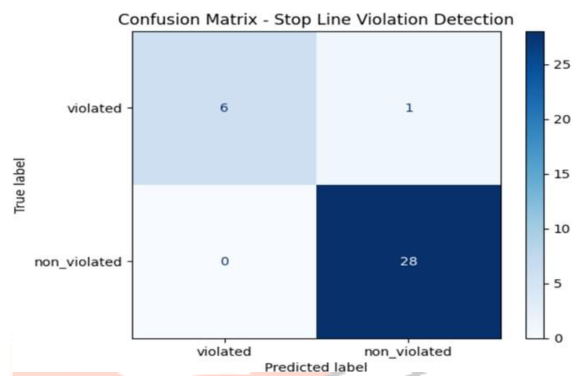


Fig. 13: Confusion Matrix for Stop-Line Crossing Detection

Performance metrics calculated from the matrix were:

- **Accuracy:** 91.11%
- **Precision:** 85.71%
- **Recall:** 85.71%
- **F1-score:** 85.71%

VII. Results and Discussion

The system was evaluated across two key violation types: helmet non-compliance and stop-line crossing. Based on test data derived from urban traffic footage, the performance was measured using confusion matrices and key classification metrics.

For the helmet detection module, the system demonstrated high precision and recall. It successfully identified 32 helmet violations and 9 correct non-violations, with only 4 false negatives and 1 false positive. This translated into an overall accuracy of 91.11%, with a strong F1-score of 92.73%. The high recall value suggests the model is effective at minimizing missed violations.

In contrast, the stop-line detection module had slightly more balanced precision and recall. It detected 6 out of 7 actual violations, along with one false positive, resulting in the same overall accuracy (91.11%) but a slightly lower F1-score of 85.71%. This outcome reflects the challenges in detecting stop-line crossing under conditions such as partial occlusion, low lighting, or high vehicle density. Fig. 14 shows the comparison of two detection modules.

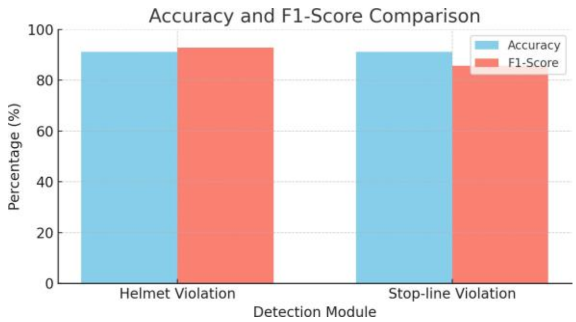


Fig. 14: Accuracy and F1-Score Comparison between Detection Modules

Overall, both modules exhibited strong detection capability and low false detection rates. The alerting mechanism via Firebase Cloud Messaging worked reliably in live tests, with officers receiving notifications in real time based on GPS proximity. Stripe-based payment verification also updated the database correctly, validating the end-to-end system flow.

These results show that the system can serve as a scalable and intelligent traffic enforcement tool for smart city infrastructure. The integration of real-time violation detection, mobile alerts, and online fine processing forms a comprehensive enforcement pipeline suitable for deployment in real-world urban environments.

VIII. Conclusion and Future Work

This work presents an integrated and intelligent traffic violation monitoring system capable of automating helmet detection and stop-line violation recognition using real-time video analysis. The use of YOLO-based object detection, combined with grayscale image processing and OCR-based ANPR, enables the system to reliably detect and document traffic offenses. Additional modules such as Firebase-based GPS tracking, FCM notification delivery, and Stripe-integrated fine collection enhance the overall functionality and practical deployment potential of the system.

Evaluation results show that the proposed system achieves a high level of accuracy, precision, and recall across both helmet and stop-line violation detection tasks. The Android application, integrated with Firebase, enables real-time field response, making it suitable for deployment by traffic enforcement agencies. Furthermore, the system supports visual analytics such as heatmaps through Google Maps API, assisting authorities in identifying high-risk areas.

For future enhancements, the system can be extended to detect additional types of violations, including red light jumping, triple riding, or mobile phone usage while driving. Incorporating night vision compatibility and thermal imaging support can further improve detection under poor lighting conditions. Additionally, integrating multilingual audio announcements using advanced speech synthesis can increase the effectiveness of in-field deterrence.

Overall, the system represents a scalable, modular solution that supports real-time traffic law enforcement and aligns with smart city infrastructure goals.

References

[1] S. Shobana and M. Jaya, "Automated Traffic Violation Monitoring System using Deep Learning," *International Journal of Computer Applications*, vol. 183, no. 42, pp. 12–18, 2021.

[2] R. Vishnupriya and M. Indumathi, "AI-based Helmet and License Plate Detection System," *International Research Journal of Engineering and Technology*, vol. 7, no. 5, pp. 2386–2390, 2020.

[3] P. Sowmya and T. Venkat, "Helmet Detection for Motorcyclists Using Image Processing," *International Journal of Engineering and Technology*, vol. 7, no. 3.6, pp. 543–546, 2018.

[4] K. Divya and B. Jansi Rani, "Traffic Rule Violation Detection Using Convolutional Neural Networks," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 6, pp. 944–950, 2019.

[5] R. Akanksha and S. Anil Kumar, "Smart Surveillance System for Traffic Violation Detection Using Raspberry Pi," *International Journal of Scientific and Research Publications*, vol. 9, no. 2, pp. 232–236, 2019.

[6] S. Sumathi and K. Jeyanthi, "Real-Time Traffic Monitoring Using YOLO and Deep Learning," *IEEE Conference on Intelligent Systems*, pp. 105–109, 2021.

[7] K. R. Harini and D. Kumar, "Helmet and Number Plate Detection Using Machine Learning," *International Journal of Computer Trends and Technology*, vol. 68, no. 6, pp. 47–50, 2020.

[8] M. Pooja and B. Kamalraj, "Machine Learning for Violation Prediction and Spatial Heatmaps," *International Journal of Engineering Research and Applications*, vol. 11, no. 4, pp. 22–27, 2021.