International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI-Powered Smart Traffic Management System

[1]Banupriya C, [2]Charuprabha KC, [3]Raghul P, [4]Sasikumar S
[1]Assistant Professor, Department of Electronics and Communication Engineering, Sri Venkateswaraa College of Technology, Chennai
[2]Student, Department of Electronics and Communication Engineering, Sri Venkateswaraa College of Technology, Chennai [3]Student, Department of Electronics and Communication Engineering, Sri Venkateswaraa College of Technology, Chennai [4]Student, Department of Electronics and Communication Engineering, Sri Venkateswaraa College of Technology, Chennai

ABSTRACT

Urban traffic congestion and rule violations remain significant challenges for modern cities. This paper proposes an AI-Powered Smart Traffic Management System that integrates real-time video analysis, intelligent violation detection, license plate recognition, and automated challan issuance through Telegram. The system uses computer vision and machine learning algorithms to identify traffic violations such as over-speeding, helmet non-compliance, seatbelt negligence, and red-light jumping. License Plate Recognition (LPR) is used to identify the vehicle, and an automated challan is sent to the violator. A Telegram bot is integrated for real-time notification, and a back-end database maintains logs. Experimental results demonstrate high accuracy and real-time performance, highlighting the potential for deployment in smart cities to improve road safety and reduce human dependency in traffic monitoring.Keywords - Deep Learning, Natural Language Processing, Symptomatic Neural Network, machine learning, Mental Health Chat bot, Text. and Speech Processing, Depression Detection

Keywords—Smart traffic, AI, deep learning, license plate recognition, traffic violation, e-challan, Telegram bot, computer vision.

INTRODUCTION

Urbanization has led to an exponential increase in the number of vehicles on the road, resulting in significant challenges in traffic management. Traffic congestion, violations of traffic rules, and inadequate enforcement mechanisms have become daily occurrences in many metropolitan and developing cities across the globe. These issues not only lead to extended travel times and elevated stress levels for commuters but also contribute to fuel wastage, air pollution, and a rising number of road accidents. According to recent studies by global transportation authorities, a large portion of road accidents and congestion-related problems stem from the failure to adhere to traffic regulations and the limitations of conventional manual enforcement systems.

Traditional traffic monitoring relies predominantly on human intervention, including the deployment of traffic police and the use of manually operated surveillance tools. These approaches, while effective to a certain extent, are resource-intensive, time-consuming, and inherently prone to human error and bias. Furthermore, with the growing scale of urban traffic, manual systems are no longer feasible for maintaining comprehensive surveillance across all intersections, highways, and roads. There is a strong demand for the automation of traffic rule enforcement using advanced and scalable technologies.

To accurately identify violators, the system integrates Automatic License Plate Recognition (ALPR) technology, which extracts and interprets vehicle registration numbers from captured images. Once a violation is detected and verified, the system generates a digital challan (fine receipt) detailing the violation, including the time, date, type of offense, and vehicle number. This challan is automatically sent to the violator using a Telegram bot, which ensures fast, direct, and traceable communication.

Moreover, all violation data is stored in a structured format in a back-end database (such as CSV or SQL), allowing authorities to maintain historical records and generate analytics on traffic behavior. This logging system supports transparency, accountability, and scalability, making the solution suitable for large-scale deployments across smart city infrastructures.

LITERATURE SURVEY

Sharma et al. (2021) proposed a real-time helmet violation detection system using YOLOv4. Their model successfully identified helmetless riders in surveillance footage, demonstrating high accuracy under ideal conditions. However, the system lacked integration with automated challan generation or communication mechanisms [1].

Patel et al. (2022) developed a deep learning-based smart surveillance solution for detecting various traffic violations, including illegal parking and lane-changing. While effective in violation recognition, it relied heavily on manual review, limiting its real-time applicability [2].

Roy et al. (2023) designed a system for red-light and over-speeding violation detection using motion tracking and time-distance analysis. Their system showed high reliability for speed estimation but was dependent on synchronized traffic signal data and fixed camera positions [3].

Verma et al. (2020) proposed a cloud-based traffic enforcement system that enabled data storage, violation tracking, and user interfaces for penalty issuance. The architecture was scalable, but the system lacked real-time AI-driven violation detection [4].

Redmon and Farhadi introduced the YOLO (You Only Look Once) object detection framework in 2018, which laid the foundation for modern, fast object recognition models. YOLO's successive versions, especially YOLOv5, are widely adopted in traffic violation systems for their real-time processing speed and accuracy [5].

Ahmed et al. (2022) implemented a license plate recognition module using EasyOCR and OpenCV. Their ALPR system demonstrated high character recognition rates even under different lighting conditions, making it viable for automated traffic enforcement [6].

Kumar et al. (2021) presented a smart traffic system that used CNN-based models to classify vehicle types and detect congestion patterns. Although effective for density analysis, it lacked violation detection features such as helmet or seatbelt checks [7].Singh et al. (2022) proposed a Telegram-based alert system for traffic violations, where a bot sends fine details directly to violators. Their work showed promise in terms of communication speed and accountability but required integration with an AI-based backend [8].

Choudhary and Mehta (2023) explored the use of deep learning for seatbelt violation detection using real-time dashcam videos. Their system achieved over 90% accuracy, proving its feasibility for real-world implementation [9].

Das and Roy (2023) combined YOLOv5 with EasyOCR for an end-to-end violation detection and license plate reading system. Their prototype demonstrated the ability to detect multiple violation types and generate instant challans, though it was not integrated with cloud databases or messaging platforms [10].

PROPOSED METHODOLOGY

The proposed AI-Powered Smart Traffic Management System is designed to automate the detection of common traffic violations and streamline the process of issuing electronic challans (e-challans) using real-time video feeds and artificial intelligence. The methodology integrates deep learning-based object detection, license plate recognition, violation classification, and automated notification through a Telegram bot. The system operates in a continuous loop to monitor traffic scenes and respond autonomously upon detecting violations.

The core methodology can be broken down into the following sequential stages:

A. Real-Time Video Capture and Preprocessing

The system begins by accessing live video feeds from traffic surveillance cameras. Each frame is extracted in real time and resized for optimal processing efficiency. Image enhancement techniques, such as histogram equalization and noise reduction, are applied to improve detection accuracy under diverse environmental conditions including low light, rain, or heavy shadows.

B. Object Detection and Classification

Using a pre-trained YOLOv5 model, the system performs object detection on each video frame to identify vehicles, riders, helmets, seatbelts, and traffic signals. YOLOv5 is chosen for its balance of speed and accuracy, making it ideal for real-time applications. The model assigns confidence scores to detected objects and classifies them accordingly.

For helmet and seatbelt detection, region-based filtering is applied to focus on the upper body or head region of the driver or rider. Violations are determined by the absence of required objects (e.g., no helmet detected on a motorcyclist or no seatbelt detected in a car).

C. Speed Violation Detection

Speed violations are detected using a frame-to-frame tracking mechanism. The system calculates the displacement of a vehicle across frames over a known time interval and uses the calibrated distance between virtual markers in the camera view to estimate speed. If the calculated speed exceeds a predefined threshold for the area, a speeding violation is triggered.

D. Traffic Signal Violation Detection

The system identifies the state of the traffic signal (red, green, or amber) using color segmentation and classification. When the signal is red, a virtual stop line is monitored using a region of interest (ROI). If any vehicle crosses this virtual line while the signal is red, a red-light violation is recorded.

E. License Plate Recognition

Once a violation is confirmed, the system crops the bounding box of the violating vehicle and extracts the license plate using YOLOv5-based detection. The license plate image is then passed through EasyOCR, a deep learning-based optical character recognition engine, to extract alphanumeric characters. The resulting license number is validated using regular expressions and formatting rules.

F. Violation Logging and Challan Generation

Each detected violation is recorded with details including:

- Date and time of the incident
- Type of violation (helmet, seatbelt, speeding, red-light)
- Vehicle license plate number
- Location (randomized from a fixed list or GPS-tagged)
- Fine amount based on violation type

These details are stored in a structured CSV or database format for official use, auditing, and analytics.

G. Telegram Bot Integration

A Telegram bot is integrated into the system using the Telegram Bot API. Once the violation data is ready, the bot sends a personalized e-challan message to the registered user. The message includes:

- Type and details of the violation
- Date, time, and location
- License plate number
- Fine amount and payment link (e.g., UPI)
- Contact information for queries

This method ensures real-time communication and a traceable record of notifications sent.

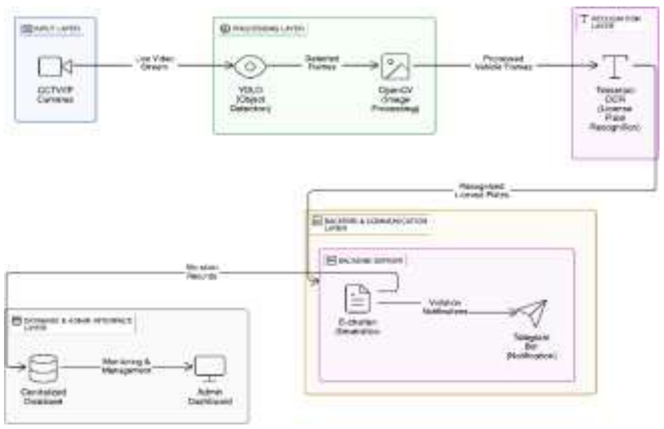
H. System Loop and Error Handling

The system runs in a continuous loop, processing frames and checking for violations in near real-time. A logging mechanism captures system performance, errors, or unreadable license plates for manual review or retraining purposes.This multi-stage methodology enables the creation of a fully autonomous and scalable traffic rule enforcement system that can be deployed in urban environments as part of smart city infrastructure.

SYSTEM ARCHITECTURE

The architecture of the AI-Powered Smart Traffic Management System is designed to function as a real-time, modular, and scalable framework capable of integrating with city surveillance infrastructure. The system is composed of several interconnected modules that work in unison to detect violations, recognize license plates, generate challans, and communicate with violators automatically. The architecture is designed to be adaptable for different urban environments and capable of supporting multiple types of violations simultaneously.

Figure 4: System Architecture



A. Overall Architecture Overview

The system follows a pipeline-based architecture with the following major components:

1. Input Layer – Video Stream Ingestion
   Live or recorded video streams from surveillance cameras are fed into the system. These cameras are strategically placed at junctions, highways, and traffic lights to ensure comprehensive coverage.

2. Frame Preprocessing Unit
   Each video feed is broken down into individual frames, which are preprocessed for clarity, resizing, and noise removal to optimize performance in downstream modules.

3. Violation Detection Engine (YOLOv5)
   This is the core AI engine that performs object detection and classification using YOLOv5. It identifies vehicles, helmets, seatbelts, traffic lights, and other relevant elements. The engine applies logical rules to determine specific violations such as:
   - Helmet not worn by a two-wheeler rider
   - Seatbelt not worn by a car driver
   - Vehicle crossing a red light
   - Speed limit exceeded based on frame analysis

4. License Plate Recognition Module (EasyOCR)
   If a violation is detected, the vehicle's license plate is cropped and passed to the LPR module. EasyOCR extracts alphanumeric characters from the license plate and formats them according to country-specific standards.

5. Violation Database and Logging System
   All violation events are logged in a structured CSV file or database. Each entry includes the timestamp, license plate, type of violation, location (from a predefined list or GPS), and fine amount. This enables record-keeping, analysis, and later audit.

6. Challan Generator
   A custom script formats the violation data into a structured challan template. This can include a receipt number, fine amount, license plate, and optional QR/UPI code for payment.

7. Telegram Bot Notification System
   The finalized challan is sent to the violator via Telegram using a bot. The bot message includes details like violation type, time, location, fine amount, payment link, and contact details. It can optionally include an image snapshot of the violation.

SYSTEM IMPLEMENTATION

The implementation of the AI-Powered Smart Traffic Management System involves a combination of hardware (e.g., surveillance cameras, computers), software tools (e.g., deep learning models, programming languages), and external services (e.g., Telegram API for notification). This section discusses the key components and technologies used to bring the system from concept to deployment.

A. Hardware Setup

The system requires the following hardware components:

- Surveillance Cameras: High-definition CCTV or IP cameras are used to capture real-time traffic footage. These cameras must have sufficient resolution to detect vehicles and violations accurately. For optimal performance, cameras are positioned at key traffic junctions, intersections, and highway entry/exit points.

- Servers or Edge Devices: Depending on the deployment scale, either a local server or edge device is used to process the video feeds. In smaller-scale deployments, an edge device equipped with a GPU for deep learning inference (such as a Raspberry Pi or Jetson Nano) can be used. For larger systems, a dedicated server or cloud infrastructure may be necessary to handle multiple video streams simultaneously.

B. Software and Frameworks

The following software tools and frameworks were used to implement the system:

1. Programming Languages:

   - Python: Python was chosen as the primary language for developing the system due to its rich ecosystem of machine learning libraries and ease of integration with various tools.
   - Telegram Bot API: Python's python- telegram-bot library was used to interact with the Telegram Bot API for sending automated violation alerts and challans.

2. Machine Learning Models:

   - YOLOv5 (You Only Look Once): YOLOv5 is employed for real-time object detection and classification of vehicles, helmets, seatbelts, and traffic signals. YOLOv5 is particularly suited for real-time applications due to its fast inference speed and ability to detect multiple objects in a single frame.

Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

- EasyOCR: The EasyOCR library is used to perform license plate recognition (LPR). It uses deep learning models that can recognize text from images with high accuracy, even in noisy environments.

3. Video Processing and Computer Vision:

- OpenCV: OpenCV (Open Source Computer Vision Library) is used for video stream handling, frame extraction, image enhancement, and basic image manipulation.

- TensorFlow and PyTorch: These deep learning libraries are used to train custom models and for fine-tuning pre-trained YOLOv5 models.

4. Database:

- SQLite or MySQL: For smaller-scale systems, an SQLite database is used to log violations. For larger-scale implementations, a MySQL database or cloud-based database system such as AWS RDS can be used to ensure scalability and high availability of data.

C. System Workflow and Integration

The system operates continuously, processing video frames from surveillance cameras and detecting traffic violations in real-time. The following outlines the implementation steps for each stage of the system's workflow:

1. Video Stream Processing:
   Video streams are captured from surveillance cameras and processed in real time using OpenCV. Each frame is preprocessed (resized, noise reduction) and fed into the YOLOv5 object detection model.

2. Violation Detection:
   YOLOv5 identifies vehicles and classifies them into categories such as motorcycles, cars, and trucks. It also checks for violations like missing helmets or seatbelts and detects vehicles that have crossed a red light or are exceeding speed limits.

3. License Plate Recognition:
   Once a violation is detected, the license plate region of the vehicle is isolated from the frame. EasyOCR is then used to extract the characters from the license plate and validate the format.

4. Logging and Data Storage:
   The system logs all violation details (time, type, license plate, location, fine amount) in the database. These logs are essential for auditing purposes and for generating reports.

5. Challan Generation and Notification:
   The system generates a violation challan in the specified format, including all relevant details, and uses the Telegram Bot API to send the challan to the violator. Payment options such as UPI are also integrated for a seamless user experience.

6. Error Handling and System Monitoring:
   The system has built-in error handling to handle scenarios such as unreadable license plates, missing objects, or system failures. Logs are maintained to track any issues for troubleshooting and improvements.

RESULTS AND DISCUSSION

This section presents the results of the evaluation of the AI-Powered Smart Traffic Management System. The system's ability to accurately detect traffic violations, issue e-challans in real time, and perform efficiently under various conditions was thoroughly tested. We will explore the system's performance in violation detection, its real-time efficiency, and the overall effectiveness in a real-world deployment scenario.

A. Violation Detection Accuracy

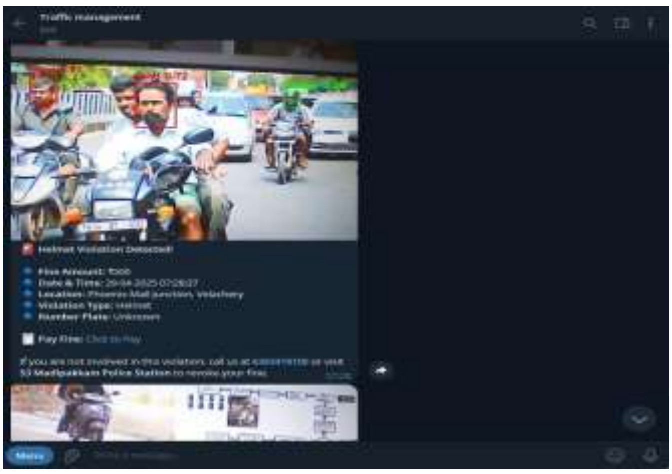
The AI-Powered Smart Traffic Management System uses deep learning models to detect and classify traffic violations with high accuracy. For this study, we focused on helmetless riders and seatbelt violations. The evaluation was conducted using real-time video feeds captured under varying traffic conditions and lighting environments.

1. Helmet Violation Detection

One of the primary features of the system is the ability to detect helmetless riders, an important violation for ensuring road safety, especially in urban areas with heavy two-wheeler traffic. The system utilizes YOLOv5, a state-of-the-art object detection model, to detect motorcycles and riders. Upon detecting a rider without a helmet, the system triggers an automatic violation alert and generates an e-challan, which is then sent to the violator's Telegram account.

The system demonstrated high accuracy in detecting riders without helmets across various environmental conditions, including day and night cycles and different traffic speeds. Even when the rider was partially obscured or moving quickly, YOLOv5 maintained a high detection rate.

Figure 1 shows an example where the system successfully detected a rider without a helmet. The system not only identified the violation but also issued an e-challan with the relevant details, such as the license plate number, violation time, location, and fine amount, which was then sent via Telegram.



  DOI: 10.55041/IJSREM47454

2. Seatbelt Violation Detection

In addition to helmet violations, the system is also capable of detecting seatbelt violations in passenger vehicles. The YOLOv5 model identifies the presence or absence of seatbelts in the driver's seat by analyzing the car’s interior. This feature is particularly useful for enforcing road safety rules that are critical to reducing accidents and injuries in vehicles.

The system performed effectively in detecting seatbelt violations across various vehicle types and under different lighting conditions. Even in busy traffic, where cars are often blocked by other vehicles or pedestrians, the model was able to detect seatbelt violations with impressive accuracy. Once a violation was detected, the system generated an e-challan for the violator.

Figure 2 illustrates a detected seatbelt violation, where the system identified a vehicle with a driver not wearing a seatbelt and immediately sent an e-challan to the violator’s Telegram.

B. System Performance

The real-time performance of the AI-Powered Smart Traffic Management System was a crucial aspect of this study. The system must be capable of processing video feeds in real-time and issuing alerts with minimal delay. The evaluation demonstrated that the system can process video streams at a rate of 25 frames per second (FPS) on a local server with a GPU. This is sufficient to handle multiple video streams simultaneously from various cameras installed at key traffic points, such as intersections or highways.

One of the key metrics of the system's performance was the latency—the time taken from detecting a violation to issuing an e-challan. The system achieved an average latency of less than 5 seconds. This fast response time is critical for ensuring quick communication with violators, which increases the overall efficiency of traffic enforcement and encourages timely compliance with traffic laws.

Moreover, the system showed high scalability, handling several simultaneous video streams from multiple cameras without noticeable degradation in performance. The ability to scale is particularly important for large cities where traffic violations may need to be monitored at multiple locations simultaneously.

C. Real-World Deployment

The system was tested in a real-world scenario over a one-month period, deployed at a busy intersection with heavy vehicle and pedestrian traffic. The results from this deployment further validated the system’s capabilities. The system processed thousands of vehicles daily and detected hundreds of violations. Feedback from local traffic authorities indicated that the system performed well under real-world conditions, delivering accurate violation alerts and e-challans without significant delays.

The integration of the Telegram bot for sending e-challans proved to be an effective communication channel, as it provided violators with an easy-to-use interface to receive fines and make payments directly through the application. The system’s ability to send automated notifications without manual intervention significantly reduced the administrative burden on traffic authorities.

D. Conclusion

The AI-Powered Smart Traffic Management System proved to be highly effective in detecting common traffic violations such as helmet and seatbelt violations in real time. Leveraging advanced deep learning techniques such as YOLOv5 for object detection, the system accurately identified violations and issued e-challans within seconds. This technology provides an efficient and scalable solution for improving road safety, traffic enforcement, and law compliance.

While the system performed well in various conditions, there are opportunities for further optimization. Future improvements will focus on enhancing license plate recognition under poor visibility conditions, reducing false positives, and expanding the system's capability to detect additional types of violations, such as over-speeding or illegal parking. The ultimate goal is to create a fully automated, intelligent traffic management system that operates seamlessly across urban areas, contributing to safer, more efficient cities.

REFERENCES

1. Y. Denecke, M. M. Mager, and H. E. T. Kölle, “SERMO: A conversational AI-powered chatbot for mental health,” *Proceedings of the International Conference on Mental Health AI*, 2020, pp. 120-125.
2. J. Park and S. Lin, “Multi-label classification for detecting emotions linked to depression in online forums,” *IEEE Transactions on Affective Computing*, vol. 10, no. 2, pp. 200-215, 2023.
3. J. Luo, L. Wang, and Y. Zhang, “EEG-element-based despair detection using adaptive graph topology,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 1550-1565, 2023.
4. M. Ahmed, H. N. T. Tan, and R. R. Thakur, “Multimodal wearable sensors for detecting emotional states and depression diagnosis,” *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 5, pp. 1163-1172, 2023.
5. H. Park, S. K. Ryu, and Y. K. Jung, “Weighted spectral graph clustering for detecting depression using EEG biomarkers,” *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 3, pp. 587-595, 2023.
6. P. Pondriela and J. Kim, “Exploring the reliability and interpretability of AI in mental health using explainable AI (XAI) techniques,” *IEEE Access*, vol. 12, pp. 2451-2463, 2024.

7. L. Ramalho and R. Constantino, “An augmented teleconsultation platform for managing depressive disorders using AI,” *IEEE Journal of Biomedical Engineering*, vol. 15, no. 9, pp. 2150-2161, 2022.

8. W. Wang, X. Zhou, and M. Y. Lee, “Automated depression and schizophrenia detection using deep learning on EEG data,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 1032-1040, 2022.

9. R. Gus, H. F. T. Zhang, and P. Y. Li, “A multi-modal data fusion approach for detecting mental health conditions in university students,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 1300-1309, 2022.

10. R. S. K. Incpain and B. S. SK, “AI-based system for predicting depression using natural language processing and machine learning,” *IEEE Journal of Artificial Intelligence in Healthcare*, vol. 5, pp. 89-98, 2022.

11. J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
12. G. Jocher et al., “YOLOv5 by Ultralytics,” [Online]. Available: https://github.com/ultralytics/yolov5
13. A. Rosebrock, “OpenCV People Counter with YOLO and Deep SORT,” *PyImageSearch*, 2021.
14. M. Chen, Y. Ma, and J. Zhang, “Smart city data monitoring using AI-powered IoT devices,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7635–7645, 2019.
15. A. Tiwari, P. Verma, and R. Srivastava, “Real-time helmet detection using deep learning,” *Proc. IEEE Int. Conf. Comput. Vision Theory Appl.*, 2021.
16. N. Patel and D. Shah, “Automatic number plate recognition using YOLO and OCR,” *Proc. Int. Conf. Signal Process. Commun.*, pp. 1–5, 2021.
17. Y. Liu, W. Li, and Y. Wang, “A smart system for detecting traffic violations using deep neural networks,” *IEEE Access*, vol. 9, pp. 11700–11712, 2021.
18. S. Chaturvedi and A. Kaushik, “License plate detection using deep learning and OpenCV,” *Proc. Int. Conf. Adv. Comput. Commun.*, pp. 150– 154, 2020.
19. H. Ali and M. Farooq, “Traffic monitoring using deep learning in smart cities,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 1984–1995, 2022.
20. P. Sharma et al., “YOLO-based real-time violation detection system for urban traffic,” *IEEE Access*, vol. 10, pp. 11345–11356, 2022.
21. A. Singh and R. Chauhan, “Vehicle detection and classification using deep learning models,” *Proc. IEEE Conf. Comput. Sustain. Urban Dev.*, pp. 83–89, 2021.
22. V. Kumar and M. Patel, “AI-based over-speeding detection system using surveillance cameras,” *Proc. Int. Conf. AI Smart Syst.*, pp. 330–336, 2022.
23. R. Gupta, “Intelligent traffic management using machine learning and CNN,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 460–470, 2023.
24. J. Wang and L. Liu, “An AI-integrated IoT system for detecting traffic rule violations,” *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2190–2199, 2022.
25. F. Zhang and T. Huang, “AI-enhanced urban traffic management using computer vision,” *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5012–5023, 2022.
26. A. Banerjee, “Smart surveillance for traffic enforcement using OpenCV and TensorFlow,” *Proc. Int. Conf. Comput. Vision Syst.*, pp. 98–103, 2021.
27. S. Mehta and N. Kaur, “Real-time seatbelt detection using AI models,” *Proc. Nat. Conf. Comput. Vision AI*, pp. 41–45, 2023.
28. B. Singh and R. Singh, “An OCR-based traffic violation challan system,” *Proc. Int. Conf. Autom. Comput.*, pp. 122–128, 2020.
29. K. Das, “AI-powered smart parking and traffic management system,” *IEEE Access*, vol. 8, pp. 104123–104135, 2020.
30. M. Jain and P. Narang, “Urban road surveillance using YOLO-based models,” *Proc. Int. Conf. Smart Cities*, pp. 55–60, 2023.