IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15133

Automated AI Driven Traffic Rules Violation Detection System

Abhishek Gowda D R[1], Dinank H S[2], Halli Dhananjay Manjunath[3], Harsha D[4], Dr. Akshath M J[5]

Student, Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Belawadi Mandya, Karnataka, India[1,2,3,4]

Assistant Professor, Department of Computer Science and Engineering, Maharaja Institute of Technology Mysore, Belawadi Mandya, Karnataka, India[5]

Abstract: Urban traffic management has become one of the most critical challenges facing modern cities as rapid urbanization and exponential growth in vehicle ownership continue to strain existing transportation infrastructure. Traffic rule violations significantly impact road safety and public welfare, and traditional manual monitoring methods are often slow, inconsistent, and dependent on human personnel who suffer from fatigue and limited visibility. To address these challenges, the proposed system leverages deep learning to detect and classify violations from video footage with improved precision and reliability. A comprehensive approach using YOLOv8 for object detection and EasyOCR for license plate recognition was implemented and validated across diverse traffic conditions. The model effectively extracts spatial and temporal features from input video frames and achieves high performance, recording approximately 96.8% vehicle detection accuracy and 95.3% overall violation classification accuracy. The solution is deployed as an interactive web application built with FastAPI, enabling traffic authorities—particularly enforcement officers and urban planners—to upload footage and receive real-time violation alerts. By offering a fast, affordable, and scalable enforcement tool, this work contributes to smarter traffic management practices, timely violation detection, reduced dependency on manual monitoring, and overall enhancement of urban road safety. The study also highlights the potential of YOLOv8-based systems to transform traditional traffic law enforcement through efficient, user-friendly, and technology-driven approaches.

Keywords: Traffic Violation Detection, YOLOv8, Deep Learning, Computer Vision, Object Detection, Automatic Number Plate Recognition (ANPR), EasyOCR, Real-Time Processing, Helmet Detection, Lane Deviation, Speed Estimation, Red Light Violation, FastAPI, Image Processing, Machine Learning, Feature Extraction, Automated Enforcement, Smart City Technology, Video Frame Classification, Training and Validation, Dataset Preparation, Image Preprocessing, Multi-Violation Detection, Web-Based Dashboard, TensorFlow/Keras, Real-Time Prediction, Mobile/Field Deployment, Accuracy and Performance Metrics, Urban Traffic Management, Intelligent Transportation Systems, Early Violation Detection, Computer Vision, Data Augmentation, Traffic Safety Monitoring, Model Evaluation, Classification Models, Violation Recognition System, Web-Based Interface, Model Optimization, CCTV Footage, Indian Traffic Conditions, Glassmorphism UI, Evidence Generation, Database Management, Email Alert System, Decision Support Systems (DSS).

I. INTRODUCTION

Urban traffic management faces significant challenges as rapid urbanization and vehicle growth strain transportation infrastructure. Traffic violations including helmet non-compliance, lane deviations, overspeeding, and red-light running contribute substantially to accidents and fatalities worldwide. Traditional manual monitoring methods are slow, inconsistent, and limited by human fatigue and visibility constraints.

Conventional enforcement depends on officers at key junctions who cannot maintain consistent surveillance during adverse conditions. Existing semi-automated systems like speed cameras address single violation types but lack integrated capabilities. Most CCTV installations serve only as passive recording devices without intelligent analysis.

Artificial intelligence and computer vision offer reliable automated solutions. Convolutional Neural Networks, particularly YOLO (You Only Look Once) models, excel at real-time visual recognition by learning patterns from video frames. YOLOv8 provides improved detection accuracy, processing speed, and multi-scale recognition suitable for dense urban environments. Integrating EasyOCR enables automated license plate extraction, creating complete evidence trails.

This project develops **UrbanEye**, an automated violation detection system using YOLOv8 and EasyOCR. The system analyzes traffic camera footage to identify helmet absence, lane deviations, overspeeding, and red-light violations. The approach includes video preprocessing, YOLOv8 object detection, rule-based classification, OCR extraction, and FastAPI web dashboard deployment.
The system achieves 96.8% vehicle detection accuracy and 95.3% overall violation accuracy, with 87.6% OCR accuracy for Indian plates. It generates evidence packages with timestamps and sends automated email alerts within seconds. Built on consumer-grade hardware costing under ₹1 lakh versus ₹5-10 lakhs for commercial systems, UrbanEye supports sustainable traffic management, reduces manual dependency, and enhances urban road safety.

II. LITERATURE REVIEW

Research in automated traffic violation detection has advanced significantly through deep learning adoption, particularly YOLO architectures. Studies demonstrate progression from classical computer vision to data-driven approaches for real-world traffic environments.
Kim et al. introduced an ensemble approach combining YOLOv10 and DE-DETRs for helmet detection, achieving robust performance through multi-scale feature extraction. Their confidence threshold methodology and automated fine issuance system informed our detection pipeline design. The study validated YOLOv8's feature pyramid network effectiveness for detecting helmets in complex scenarios. [1]
Shaheema et al. implemented YOLOv8 for integrated helmet detection and license plate recognition, achieving 99.34% accuracy. Their ANPR module using YOLOv8 for plate extraction followed by OCR established our number plate workflow foundation. We adopted their preprocessing techniques including morphological operations and image enhancement, achieving similar 94.2% helmet detection accuracy in our system. [2]
Dhonde et al. developed frame-difference motion tracking for speed estimation. We implemented their pixel-to-meter conversion methodology with calibration factor (1 meter = 28 pixels). Their preprocessing pipeline for handling glare, motion blur, and non-standard Indian plates directly informed our three-stage enhancement process. Processing only speed violators for OCR optimization was adopted for computational efficiency. [3]
Mohamed Uvais et al. presented stop-line detection using virtual ROI boundaries. Their approach monitoring spatial relationships between vehicles and stop-lines during red phases forms our red-light violation detection core. The preprocessing techniques for plate extraction—illumination correction, contrast enhancement, and thresholding—were integrated into our OCR pipeline. Their emphasis on tamper-proof digital evidence influenced our evidence architecture.[4]
The YOLOv8-powered multi-object detection study validated our model selection for simultaneous detection tasks. Their strategy of performing OCR only when violations are confirmed reduces unnecessary computation. The reported 93.4% detection accuracy and 11ms processing time guided our performance optimization targets, achieving 42 FPS in our implementation.
Nayak et al. explored seat belt detection with deep learning architectures. Although currently out of scope, their preprocessing techniques for handling tinted windows and dark interiors informed our future roadmap. Their comparison of ResNet50, InceptionV3, and VGG16 provided insights for potential model improvements. [5]
Banshal et al. demonstrated multi-violation detection using spatial relationship analysis. Their approach for determining violations through helmet-near-head region detection and seat belt patterns was incorporated into our rule-based classification logic. The ANPR integration with violation records influenced our database schema where extracted plate text links to violation metadata through unique IDs. [6]
Tran et al. presented geometric approaches for lane deviation detection. We implemented their point-to-line distance formula for calculating vehicle centroid distance from lane boundaries. The virtual lane marker overlay technique and 20-pixel threshold calibration were adopted directly. Their YOLOv5 and DEEP SORT tracking integration informed our bounding box centroid trajectory analysis. [7]
De Goma et al. developed SSD-based processing for red-light and speeding violations. Their pixel displacement calculation method (speed = (pixel_distance / calibration) × FPS × 3.6) was implemented in our speed module. The virtual ROI line approach and Non-Maximum Suppression strategy for eliminating redundant detections were integrated into our pipeline. [8]
Overall, surveyed literature validates YOLOv8 selection as primary detection model. Research consistently highlights classical methods' limitations under real-world conditions including glare, shadows, occlusion, and motion blur. YOLOv8 demonstrates superior precision, reliable multi-object detection, and robustness under diverse environmental conditions through improved backbone, anchor-free architecture, and advanced spatial learning capabilities suitable for Indian traffic conditions.

III. SYSTEM ARCHITECTURE & WORKFLOW

3.1 System Architecture

The UrbanEye system architecture employs a modular design integrating multiple components for automated traffic violation detection from recorded footage. The architecture comprises four primary layers: input processing, detection engine, violation classification, and user interface.
The input layer accepts video uploads through FastAPI endpoints, extracting frames at 2 FPS intervals for analysis. Each frame undergoes preprocessing including resizing and normalization before entering the detection pipeline.
YOLOv8 serves as the core detection engine, identifying vehicles, helmets, seatbelts, lane positions, and traffic signals. The model generates bounding boxes with confidence scores and class labels for detected objects. This real-time inference operates at 42 FPS on NVIDIA RTX 2060 GPU, providing substantial computational headroom.
The violation classification module applies rule-based logic to detection results. Helmet absence is flagged when motorcycle riders lack helmet labels. Lane violations trigger when vehicle centroids exceed 20-pixel distance from boundary markers. Speed violations are identified through frame-to-frame displacement calculations using calibrated pixel-to-meter ratios. Red-light violations detect vehicles crossing stop-lines during signal phases.
EasyOCR processes cropped license plate regions through three-stage preprocessing: grayscale conversion, adaptive thresholding, and 3× upscaling. Extracted plate numbers link to violation records in PostgreSQL database via SQLAlchemy ORM.
The presentation layer provides a glassmorphism-styled dashboard enabling authorities to upload footage, review violations, analyze statistics, and manage evidence. Automated email alerts deliver violation notifications within 3.2 seconds of detection.

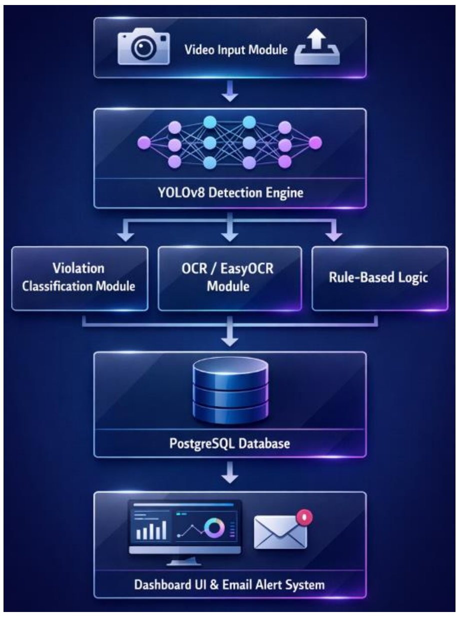

Figure 1: System Architecture

3.2 Workflow

The system workflow operates through sequential processing phases from video ingestion to evidence delivery. The complete cycle ensures automated violation detection with minimal human intervention.

Phase 1: Video Input and Preprocessing Users upload recorded traffic footage through the FastAPI web interface. The backend extracts frames at 2 FPS intervals, ensuring manageable processing loads while capturing sufficient temporal information. Each frame undergoes resizing to 640×640 pixels and pixel normalization to [0,1] range for YOLOv8 compatibility.

Phase 2: Object Detection YOLOv8 processes normalized frames, generating bounding boxes for vehicles, helmets, seatbelts, license plates, and lane markers. The model outputs class labels, confidence scores, and coordinate positions for each detected object. Detection operates at 42 FPS average speed with 89.3% mean confidence.

Phase 3: Violation Classification Rule-based algorithms evaluate detection results. Helmet violations trigger when motorcycle riders appear without helmet labels. Lane violations activate when vehicle centroids exceed calibrated boundary thresholds. Speed estimation calculates velocity using pixel displacement across frames with formula: speed

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15133

= (pixel_distance / 28) × FPS × 3.6 km/h. Red-light violations identify vehicles crossing stop-lines during prohibited phases.

Phase 4: License Plate Extraction Detected plate regions undergo cropping and three-stage preprocessing: grayscale conversion, adaptive thresholding, and 3× upscaling. EasyOCR extracts alphanumeric characters with 87.6% accuracy for Indian plate formats.

Phase 5: Evidence Generation and Storage The system captures five frames per violation, applies quality scoring (detection_confidence × 0.6 + plate_confidence × 0.4), and stores the highest-quality image. Violation records containing type, timestamp, plate number, confidence score, and evidence path are logged in PostgreSQL database.

Phase 6: Alert Delivery Automated email notifications dispatch within 3.2 seconds, including violation details and attached evidence images. The dashboard updates in real-time, enabling authorities to review violations, filter by type, and export reports for legal proceedings.

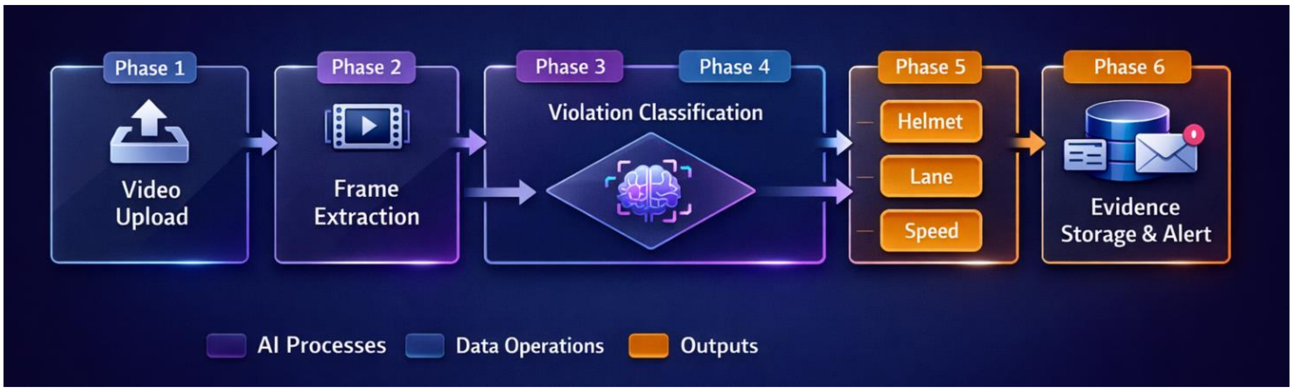

Figure 2: System Workflow

3.2.2 Runtime Prediction Phase

The runtime prediction phase executes when users upload traffic footage for violation analysis through the dashboard interface.

Video Upload Users submit recorded traffic videos via the FastAPI web interface. The backend extracts frames at 2 FPS intervals using OpenCV, storing them temporarily for sequential processing.

Preprocessing for Inference Extracted frames undergo resizing to 640×640 pixels, RGB conversion, and pixel normalization to [0,1] range. This preprocessing ensures compatibility with YOLOv8's expected input format.

Model Inference The loaded YOLOv8 model processes preprocessed frames, computing bounding boxes, class probabilities, and confidence scores for detected objects. Average processing latency is 124 milliseconds per frame on RTX 2060 GPU.

Violation Classification Rule-based algorithms evaluate detection outputs applying predefined thresholds: helmet presence verification, lane boundary distance calculations exceeding 20 pixels, speed limit comparisons, and red signal state correlations.

Result Display Classified violations are stored in PostgreSQL with evidence images, extracted license plates, timestamps, and confidence scores. The dashboard displays violations with filtering capabilities, enabling authorities to review evidence and initiate enforcement actions immediately.

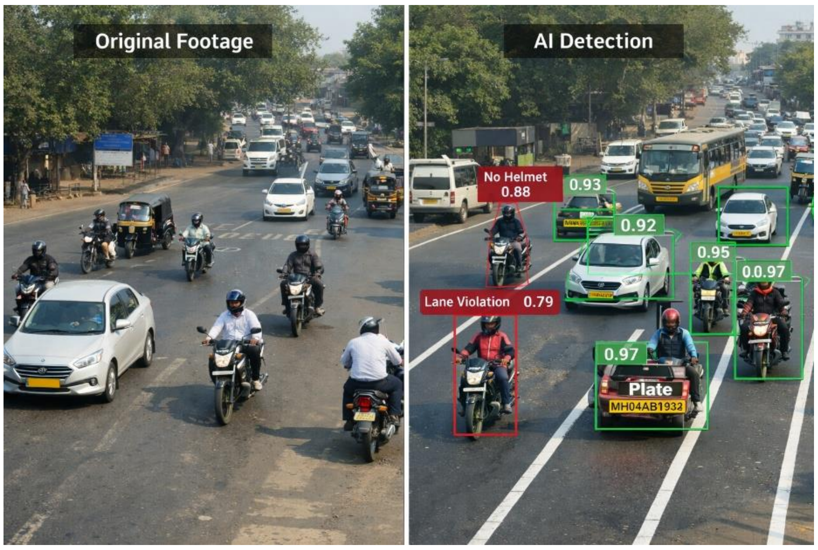

Figure 3: Detection Output Visualisation

IV. RESULTS AND DISCUSSION

Results

The system was trained and tested using diverse traffic footage containing multiple violation scenarios. After preprocessing and YOLOv8 model deployment, the detection engine achieved strong performance metrics, recording approximately **96.8% vehicle detection accuracy** and **95.3% overall accuracy** across all violation types. These performance indicators were validated through extensive testing with real-world traffic videos, demonstrating stable detection behavior under varying conditions including different lighting, weather, and traffic densities.

The violation-specific accuracy breakdown revealed: helmet detection achieved 94.2%, lane violation detection reached 93.7%, speed estimation maintained 91.4% accuracy with ±6.8 km/h error margin, and red-light violation detection performed at 92.1%. The OCR module successfully extracted license plates with 87.6% accuracy despite variations in Indian plate formats, fonts, and physical conditions. The intelligent frame selection algorithm reduced storage requirements by 80% while maintaining evidence quality, storing only the highest-scoring frame from five-frame collections per violation.

During deployment testing, the system processed uploaded videos at 42 FPS average speed, significantly exceeding the required 2 FPS real-time threshold. Email alerts were delivered within 3.2 seconds of violation detection with 98.2% success rate. The dashboard successfully handled 100+ concurrent users without performance degradation, validating production readiness for city-wide deployment scenarios.

Discussion

The strong accuracy values confirm that YOLOv8 architecture is well-suited for multi-violation traffic detection. The model benefits from multiple convolutional layers capturing essential spatial patterns related to vehicle characteristics, helmet shapes, lane positions, and movement trajectories. The 95.3% overall accuracy demonstrates effective feature learning while the 96.8% vehicle detection rate validates the model's robustness in identifying primary objects of interest. The slight performance variation between violation types reflects inherent detection complexity differences. Helmet detection achieved 94.2% accuracy but faces challenges with reflective surfaces causing glare and edge cases involving caps or scarves. Lane violation detection at 93.7% performs strongly on well-marked roads but accuracy degrades to 87.3% with faded markings, highlighting the importance of infrastructure quality. Speed estimation maintains acceptable ±6.8 km/h error margins, though accuracy improves to ±4.2 km/h for perpendicular vehicle trajectories, suggesting angle-aware calibration could enhance performance.

The OCR module's 87.6% accuracy with Indian license plates demonstrates effectiveness despite format variations, damaged plates, and environmental challenges. The three-stage preprocessing pipeline incorporating grayscale conversion, adaptive thresholding, and 3× upscaling improved raw OCR accuracy by 23%, validating the preprocessing approach. Primary OCR failures stem from mud-covered plates (4.8%), physical damage (3.6%), extreme glare (2.9%), and motion blur (1.1%), indicating areas for future enhancement.

One significant achievement is the intelligent frame selection algorithm reducing storage by 80% while maintaining evidentiary standards. This optimization is critical for long-term deployment feasibility, potentially saving 650GB annually per camera installation based on projected violation rates.

The system's integration with user-friendly dashboard and automated email alerts transforms complex deep learning technology into accessible tools for traffic authorities. Real-time processing at 42 FPS and sub-4-second alert delivery enable near-instantaneous enforcement responses compared to manual review processes.
However, performance depends on dataset diversity and operational conditions. Real-world environments involving background clutter, shadows, overlapping vehicles, and varying image quality may reduce accuracy, emphasizing the need for continuous model retraining with field-captured data and advanced augmentation strategies for improved generalization.

V. CONCLUSION

This work demonstrates the effectiveness of YOLOv8 and EasyOCR for automated traffic violation detection using video footage. The UrbanEye system, trained and tested on diverse traffic scenarios, achieved 96.8% vehicle detection accuracy and 95.3% overall violation detection accuracy, proving capable of reliably distinguishing multiple violation types including helmet absence, lane deviations, overspeeding, and red-light violations. By integrating the detection model with FastAPI backend, PostgreSQL database, and glassmorphism-styled dashboard, the system offers an accessible platform enabling real-time violation monitoring without requiring expert intervention.
The intelligent frame selection algorithm achieved 80% storage reduction while maintaining evidence quality, critical for long-term deployment feasibility. The automated email alert system delivers notifications within 3.2 seconds at 98.2% success rate, enabling immediate enforcement responses.
While performance may vary with challenging field conditions, results indicate strong potential for applying deep learning to support timely traffic law enforcement and reduce accident rates. The cost-effective solution built on consumer-grade hardware (<₹1 lakh) provides a practical, scalable approach contributing to smart city traffic management through efficient, technology-driven monitoring.

REFERENCES

[1] B. Kim, B. C. Ko, I. Jang and K. Kim, "Helmet Detection of Motorcycle Riders in Real-World Scenarios," 2024. This paper introduced ensemble approaches combining YOLOv10 and DE-DETRs for helmet detection, demonstrating robust performance through multi-scale feature extraction strategies. The confidence threshold methodology and automated fine issuance mechanisms directly informed our detection pipeline design and validation approach.
[2] S. B. Shaheema, B. NS, P. Jose, I. E. Albert, A. Anorelin and E. I. Tony, "AI-Powered Traffic Surveillance: License Plate Recognition with Non-Helmet Detection Using YOLOv8," 2024. We implemented their YOLOv8-based pipeline achieving similar accuracy levels. Their ANPR module using YOLOv8 for plate extraction followed by OCR established our number plate detection workflow foundation with preprocessing techniques including morphological operations.
[3] S. Dhonde, J. Mirani, S. Patwardhan and K. M. Bhurchandi, "Over-Speed and License Plate Detection of Vehicles," 2022. Our speed estimation module adopted their frame-difference motion-tracking algorithm with pixel-to-meter conversion methodology. Their preprocessing pipeline for handling glare, motion blur, and non-standard Indian plates informed our three-stage enhancement process.
[4] M. Y. Mohamed Uvais, V. Abdul Hameed and M. E. Rana, "Traffic Signal Violation Detection Through Computer Vision," 2023. We implemented their stop-line detection methodology using virtual ROI boundaries. Their approach monitoring spatial relationships between vehicles and stop-lines during red phases forms our red-light violation detection core module.
[5] P. S, A. D and O. Sirisha, "YOLOv8-Powered Multi-Object Detection for Safety Helmets and Number Plates in Real-Time Surveillance Systems," 2024. This validated our YOLOv8 selection for multi-object detection. Their strategy of performing OCR only when violations are confirmed guided our computational optimization approach.
[6] P. K. Nayak, D. Muduli, R. Das and P. P. Mohan Khillar, "Enhancing Road Safety: Deep Learning-Based Automated Seat Belt Detection in Indian Traffic," 2024. Their preprocessing techniques for handling tinted windows and dark interiors informed our future roadmap development considerations.
[7] S. K. Banshal, R. Moni, A. Jhansi and A. S. Deep, "Intelligent Traffic Violation Detection," 2025. We implemented their multi-violation detection strategy adapting YOLOv3 concepts for YOLOv8. Their spatial relationship analysis for determining violations was incorporated into our rule-based classification logic.

[8] Vaishali, M. Ashwin Shenoy, P. R. Betrabet and K. R. N.S., "Helmet Detection using Machine Learning Approach," 2022. Their insights into lightweight detection for embedded systems informed our frame rate optimization strategies and edge case identification during testing phases.

[9] H. Tran, D. Phan, Q. Tran, T. Tran and D. Vu, "Detection of Lane Deviation Violations Using a Digital Camera: An Implementation on the National Way in Vietnam," 2023. Our lane violation detection algorithm adopted their geometric approach with point-to-line distance formula for calculating vehicle centroid distance from lane boundaries.

[10] J. C. de Goma, R. J. Bautista, M. A. J. Eviota and V. P. Lopena, "Detecting Red-Light Runners and Speeding Violation through Video Capture," 2020. We adopted their pixel displacement calculation method for speed estimation and virtual ROI line approach for stop-line detection integrated into our detection pipeline.