International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

AI-Powered Traffic Violation Detection and Evidence Generation System

V.Sunil Anandh,M.E*, Nithish Kumaar S**, Mani Matheswaran M***, Vignesh K****, Sherine Samuel*****

*(*Assistant Professor, Department of Computer Science and Engineering, Loyola Institute of Tech. & Science sunilanandhvm@gmail.com)*

**(*Department of Computer Science and Engineering, Loyola Institute of Tech. & Science nknithish858585@gmail.com)*

***(*Department of Computer Science and Engineering, Loyola Institute of Tech. & Science matheswaran1526@gmail.com)*

****(*Department of Computer Science and Engineering, Loyola Institute of Tech. & Science vignesh8872@gmail.com)*

*****(*Department of Computer Science and Engineering, Loyola Institute of Tech. & Science petszone987@gmail.com)*

Abstract

Road traffic violations such as helmet-rule violations, triple-riding, red-light jumping, and wrong-lane movement continue to increase the burden on traffic management agencies, particularly in densely populated urban regions. Manual monitoring of surveillance feeds is labor-intensive, difficult to scale, and often inconsistent under continuous operation. This manuscript presents an AI-powered traffic violation detection and evidence generation system designed to automatically analyze CCTV streams, identify rule violations in real time, and generate legally useful digital evidence records. The proposed framework integrates a deep-learning-based detector, multi-object tracking, automatic number plate recognition (ANPR), and rule-based event reasoning in a unified pipeline. The system is intended for deployment on both edge devices such as NVIDIA Jetson platforms and GPU-enabled traffic control servers. A major focus of the work is suitability for Indian road conditions, where dense traffic, regional license plate variations, inconsistent lane discipline, and mixed illumination create major challenges for visual analytics. The manuscript contributes a structured system architecture, formal event-scoring equations, deployment-oriented design choices, and a journal-ready methodological presentation aligned with the concept introduced in the project review presentation. The proposed model supports automated evidence packs consisting of image snapshots, violation labels, timestamp, recognized license plate text, and camera or location metadata. Such a framework can improve enforcement consistency, reduce dependence on manual observation, and support scalable smart-city traffic monitoring.

Keywords: intelligent transportation systems, traffic violation detection, evidence generation, YOLOv8, multi-object tracking, ANPR, OCR, edge AI

INTRODUCTION

Traffic surveillance is an essential component of modern smart-city infrastructure. However, in many practical deployments, surveillance systems are still used only as passive recording tools rather than as intelligent enforcement assistants. Human operators are expected to watch multiple camera streams, notice rule violations, identify offending vehicles, and prepare actionable evidence manually. This workflow is inefficient, error-prone, and unsuitable for large-scale deployment.

Recent developments in deep learning, especially in real-time object detection and tracking, have made it possible to automate complex perception tasks in unconstrained traffic environments. Modern detectors can identify motorcycles, cars, riders, helmets, and number plates with high speed, while tracking algorithms maintain the identity of moving objects across frames. Combined with OCR engines, these techniques can form the basis of end-to-end automated traffic enforcement systems.

Most existing studies, however, focus on isolated tasks such as helmet detection, seat-belt recognition, signal violation detection, or license plate recognition. Few systems integrate all these capabilities into one practical workflow that not only detects a violation but also preserves the identity of the violator and produces a structured evidence package. This limitation is even more significant in Indian traffic scenarios, where heavy occlusion, regional number plate diversity, night-time glare, and dense mixed traffic complicate automated monitoring.

This manuscript proposes a unified AI-powered traffic violation detection and evidence generation system for real-time CCTV analytics. The system is designed to detect multiple classes of violations, maintain vehicle-level temporal consistency, extract number plates, and produce evidence records suitable for dashboard review and future enforcement integration.

The main contributions of this work are as follows:

1. A unified multi-violation framework covering no-helmet riding, triple-riding, red-light jumping, wrong-lane movement, and extensible future rules.
2. A deployment-aware architecture combining detection, tracking, ANPR, OCR, and evidence synthesis in one pipeline.
3. Formalized event-scoring equations to improve decision consistency and reduce spurious alerts.
4. An edge-friendly design strategy targeting Jetson devices and mid-range GPU systems for practical field deployment.

MATERIALS AND METHODS

1. Problem Definition

Let a traffic video stream be represented as a sequence of frames:

$$I = \{I_1, I_2, \ldots, I_T\}$$

where I_t denotes the frame captured at time instant t. The goal of the proposed system is to map the incoming frame sequence to a set of verified violation events:

$$V = \{v_1, v_2, \ldots, v_K\}$$

where each event v_k contains a vehicle identity, violation label, supporting snapshot, timestamp, plate text, and location metadata. The task is therefore formulated as a structured video understanding problem involving detection, tracking, rule reasoning, and evidence synthesis.

2. System Overview

The proposed framework processes video captured by fixed traffic cameras and transforms the video stream into structured violation events. The processing pipeline consists of six major stages:

1. Video acquisition and frame preprocessing.
2. Multi-class object detection.
3. Multi-object tracking.
4. Violation reasoning and temporal verification.
5. ANPR and OCR.
6. Evidence generation, storage, and dashboard integration.

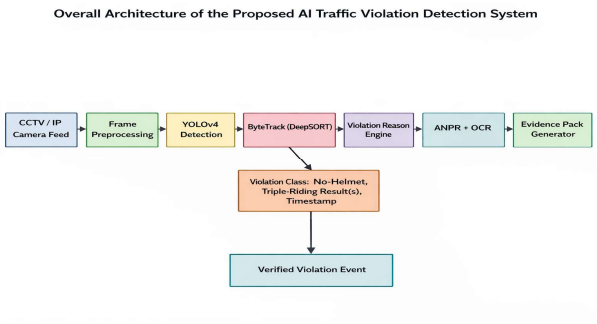

Figure 1. Overall architecture of the proposed traffic violation detection and evidence generation framework

3. Detection Model

The object detection layer is built around a YOLO-family architecture, with YOLOv8 selected as the base detector due to its balance between accuracy and real-time throughput. The detector is configured as a multi-class model capable of identifying:

- motorcycles
- riders
- helmets
- cars
- buses
- trucks
- number plates
- traffic signal state regions where required

The output of the detector for frame t is represented as:

$$D_t = \{(b_i^t, c_i^t, s_i^t)\ |\ i = 1,2,\ldots,N_t\}$$

where b_i^t denotes the bounding box, c_i^t denotes the predicted class label, and $s_i^t \in [0,1]$ denotes the detection confidence for the i^{th} object in frame t.

4. Tracking Module

Since single-frame detection is insufficient for robust traffic enforcement, a multi-object tracker is used to preserve temporal identity. ByteTrack is preferred for its efficiency in crowded scenes, while DeepSORT remains a viable alternative when appearance embeddings are required. The tracking module associates detections between consecutive frames using motion and overlap cues.

The intersection-over-union (IoU) between two bounding boxes A and B is given by:

$$\operatorname{IoU}(A,B)=\frac{\operatorname{area}(A \cap B)}{\operatorname{area}(A \cup B)}$$

For each object track τ_k, the system stores a trajectory:

$$\tau_k = \{b_k^{t_s}, b_k^{t_s+1}, \ldots, b_k^{t_e}\}$$

where t_s and t_e are the start and end frames of track k. During association, each detection is matched to the active track that maximizes the combined association score:

$$A_{ij} = \lambda_1 \operatorname{IoU}(b_i^t,\hat{b}_j^t) + \lambda_2 \exp\left(-\frac{\|p_i^t-\hat{p}_j^t\|_2^2}{2\sigma^2}\right)$$

where \(\hat{b}_j^t \) is the predicted location of track \(j \), \(p_i^t \) is the center of the current detection, and \(\lambda_1 + \lambda_2 = 1 \).

5. Violation Reasoning Logic

The violation engine combines detector outputs, track continuity, and scene constraints to infer rule violations. A violation is not declared from a single frame alone. Instead, the system aggregates evidence over a short temporal window to reduce false positives.

Let \(V_j \) denote a violation class. The aggregated violation score is defined as:

$$S_j = \alpha C_{det,j} + \beta C_{trk,j} + \gamma C_{ocr,j} + \delta C_{ctx,j}$$

subject to:

$$\alpha + \beta + \gamma + \delta = 1$$

where:

- \(C_{det,j} \) is the average object-detection confidence relevant to violation class \(j \)
- \(C_{trk,j} \) is the temporal tracking consistency score
- \(C_{ocr,j} \) is the normalized OCR confidence for the offending vehicle plate
- \(C_{ctx,j} \) is the rule-specific contextual score derived from lane, signal, or rider-count logic

A violation is confirmed when:

$$S_j \ge \theta_j$$

where \(\theta_j \) is the decision threshold for violation class \(j \).

Violation class	Primary visual cues	Decision condition
No-helmet	motorcycle, rider head region, missing helmet	helmet absent over temporal window
Triple-riding	motorcycle and rider count	rider count greater than two across consecutive frames
Red-light jump	vehicle track, stop line, signal phase	vehicle crosses stop line while signal state is red
Wrong-lane	lane polygon, track direction	motion direction inconsistent with authorized lane direction

Table 1. Rule definitions used in the violation reasoning engine

5.1 No-Helmet Violation

If a motorcycle track contains one or more rider detections and no valid helmet detection overlaps with the rider head region for a predefined temporal window, a no-helmet event is triggered.

5.2 Triple-Riding Violation

If the number of rider detections associated with a motorcycle exceeds two for more than \(n \) consecutive frames, the event is classified as triple-riding.

$$R_k = \frac{1}{n}\sum_{t=t_0}^{t_0+n-1} r_k^t$$

$$R_k > 2 \Rightarrow \text{triple-riding violation}$$

where \(r_k^t \) is the number of riders associated with motorcycle track \(k \) in frame \(t \).

5.3 Red-Light Jumping

For signalized intersections, a stop-line crossing event is treated as a red-light violation only if the vehicle crosses the virtual stop line during the red phase.

$$L_k = y_k^{t_2} - y_k^{t_1}$$

$$\big(y_k^{t_1} < y_{stop} \ \land\ y_k^{t_2} \ge y_{stop} \ \land\ \phi(t_2)=1\big) \Rightarrow \text{red-light violation}$$

where y_{stop} is the stop-line position and $\phi(t)=1$ indicates that the signal is red at time t.

5.4 Wrong-Lane Movement

Lane polygons are defined during camera calibration. If the dominant motion vector of a vehicle track violates the permitted direction of its lane region, a wrong-lane flag is raised.

The directional consistency score may be expressed as:

$$\rho_k = \frac{u_{k,x}d_{\ell,x} + u_{k,y}d_{\ell,y}}{\sqrt{u_{k,x}^2 + u_{k,y}^2}\sqrt{d_{\ell,x}^2 + d_{\ell,y}^2}}$$

where $\vec{u}_k$ is the observed motion vector of track k and $\vec{d}_\ell$ is the permitted lane direction vector. A wrong-lane event is raised when $\rho_k < 0$ or falls below an application-specific threshold.

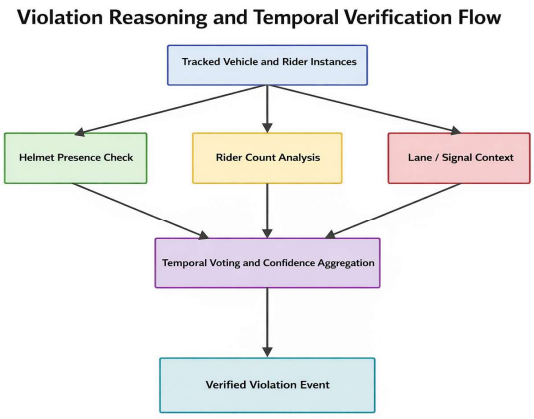

Figure 2. Violation reasoning and temporal verification flow

6. Algorithmic Workflow

The proposed event generation procedure is summarized below.

Algorithm 1. Multi-violation event generation procedure

Step	Operation
1	Acquire video frame I_t from traffic camera
2	Apply preprocessing for resizing, denoising, and contrast normalization
3	Run YOLOv8 detector to obtain object and plate candidates
4	Associate detections across frames using ByteTrack or DeepSORT
5	Group rider, helmet, and motorcycle instances by track identity
6	Evaluate rule logic for no-helmet, triple-riding, red-light, and wrong-lane violations
7	Verify persistence of the event across a temporal window
8	Crop number plate region and perform OCR
9	Validate OCR string using Indian plate format rules
10	Generate evidence pack and store it in the database for dashboard review

7. ANPR and OCR Module

After a violation is verified, the system crops the number plate region and passes it through ANPR and OCR stages. The OCR output is then post-processed using a regular expression tailored to Indian vehicle registration formats.

The recognized plate string is represented as:

$$P^{*} = \arg\max_{P \in \Omega} \Pr(P \mid I_{plate})$$

The average OCR confidence is:

$$C_{ocr} = \frac{1}{m}\sum_{q=1}^{m} c_q$$

where Ω is the set of valid Indian registration patterns and c_q is the

confidence assigned to the q^{th} recognized character.

8. Evidence Pack Generation

The evidence generator stores the most informative frames associated with each violation track. Each evidence pack E_k is defined as:

$$E_k = \{F_k, P_k^{*}, t_k, g_k, cam_k, v_k, S_k\}$$

where:

- F_k is the selected evidence frame
- P_k^{*} is the validated plate text
- t_k is the timestamp
- g_k is the GPS coordinate or location tag
- cam_k is the camera identifier
- v_k is the violation label
- S_k is the violation confidence score

This packaging step is critical because the practical value of the system depends not just on detection but on producing structured, reviewable, and auditable evidence.

The final evidence frame is selected by maximizing a composite quality score:

$$Q_f = \eta_1 s_{plate,f} + \eta_2 s_{sharp,f} + \eta_3 s_{vis,f}$$

where $s_{plate,f}$ is plate confidence, $s_{sharp,f}$ is image sharpness, $s_{vis,f}$ is visibility of the violating subject, and $\eta_1+\eta_2+\eta_3=1$.

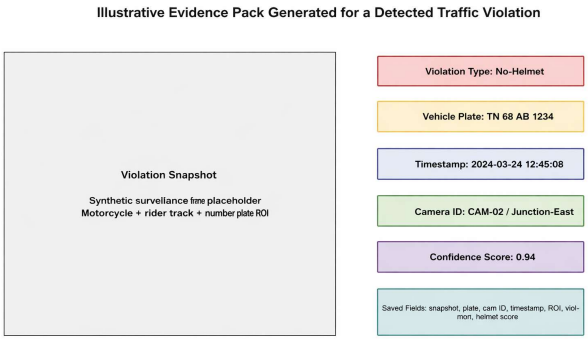

Figure 3. Illustrative evidence pack generated for a detected traffic violation

9. Dataset Design and Annotation Strategy

Since the proposed work targets Indian road conditions, the dataset should include diverse scenes such as urban roads, signalized junctions, highways, narrow streets, and mixed day-night conditions. Annotation must capture both object-level and event-level semantics.

Annotation category	Labels / attributes
Vehicle objects	motorcycle, car, bus, truck, auto-rickshaw
Rider attributes	rider, pillion rider, helmet, no-helmet
Plate information	plate bounding box, visible/not visible, readable/not readable
Violation metadata	no-helmet, triple-riding, wrong-lane, red-light jump
Scene metadata	time of day, rain/fog, traffic density, camera angle

Table 2. Recommended annotation schema for the proposed system

10. Implementation Environment

The deployment environment is designed to support both prototyping and real-world inference.

Category	Recommended specification
Camera	1080p IP camera with night vision and WDR

Edge device	NVIDIA Jetson Xavier NX / Jetson Orin
Server option	Desktop with RTX 3060 / 4060 GPU
CPU / RAM	Intel i7 or Ryzen 7, 16 GB RAM or higher
Storage	512 GB SSD + 1 TB archival storage
Frameworks	Python 3.10+, PyTorch, OpenCV
Detection	YOLOv8
Tracking	ByteTrack or DeepSORT
OCR	PaddleOCR or EasyOCR
Acceleration	TensorRT or ONNX Runtime
Backend	Flask or FastAPI
Frontend	React dashboard

Table 3. Hardware and software configuration

11. Training and Deployment Strategy

The training pipeline should include annotation validation, class balancing, data augmentation, and staged validation on both daylight and night-time samples. Augmentation may include blur simulation, brightness variation, rain-like distortion, motion streaks, and partial occlusion to improve robustness. For deployment, the trained detector can be exported to ONNX and optimized through TensorRT for edge inference on Jetson-class hardware.

From a systems perspective, model execution should follow an asynchronous design in which frame capture, inference, OCR, and evidence storage are handled in separate queues or threads. This reduces blocking latency and improves throughput under high camera load.

12. Proposed Evaluation Metrics

The final implementation should be evaluated using both perception and enforcement metrics.

Module	Metric
Object detection	precision, recall, mAP@0.5, mAP@0.5:0.95
Tracking	IDF1, MOTA, MOTP, identity switches
ANPR	plate detection accuracy, OCR accuracy, exact plate match rate
Violation detection	event precision, event recall, F1-score
Deployment	frames per second, latency per frame, memory footprint

Table 4. Performance metrics planned for system evaluation

The F1-score for event detection is computed as:

$$F\_1 = \frac{2PR}{P + R}$$

where \(P \) is precision and \(R \) is recall.

To convert model-level predictions into a deployment-oriented quality indicator, a normalized system performance index may be defined as:

$$\Pi = \omega\_1 P\_e + \omega\_2 R\_e + \omega\_3 T\_a + \omega\_4 O\_a$$

where \(P_e \) is event precision, \(R_e \) is event recall, \(T_a \) is normalized throughput adequacy, \(O_a \) is OCR adequacy, and \(\omega_1+\omega_2+\omega_3+\omega_4=1 \).

Indicator	Target interpretation	Target value
Event precision	correct violation alarms among all alarms	at least 0.90
End-to-end latency	time from frame capture to evidence-ready event	below 100 ms per frame on supported hardware
Plate readability success	proportion of verified events yielding valid plate text	at least 0.85 under daytime conditions
Tracking continuity	stable identity retention through short occlusions	at least 0.80 normalized stability

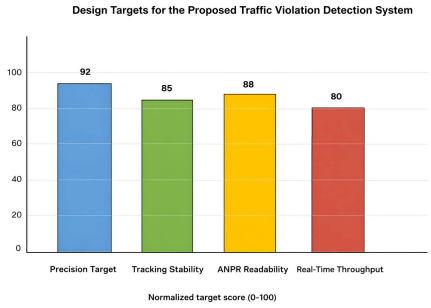

Figure 4. Design-target performance graph for the proposed system

13. Failure Cases and Mitigation Strategy

To improve deployment readiness, the system should explicitly account for challenging scenarios.

Failure scenario	Likely impact	Mitigation approach
Helmet partially occluded	false no-helmet alarms	temporal voting and rider-head association refinement
Plate blur at night	ANPR failure	multi-frame plate super-resolution or best-frame selection
Dense intersection crowding	ID switches in tracking	higher frame rate capture and track confidence gating
Signal reflection or glare	red-light reasoning errors	signal ROI calibration and phase smoothing
Camera perspective drift	wrong-lane false positives	periodic recalibration and lane polygon validation

Table 6. Anticipated failure cases and mitigation mechanisms

RESULTS AND DISCUSSION

The present manuscript is derived from a project review concept note and system design presentation. Therefore, this section is framed as a journal-style analytical discussion of expected system behavior and evaluation readiness rather than a fabricated experimental report. Numerical results should be inserted only after model training, dataset curation, and field validation are completed.

1. Expected Performance Trends

The integration of YOLOv8 with ByteTrack or DeepSORT is expected to provide strong suitability for real-time traffic analytics because the detection layer supplies spatial semantics while the tracking layer maintains temporal continuity. This combination is particularly important in violation monitoring because brief detector failures in a single frame should not immediately cancel a genuine event.

The use of temporal verification is also expected to improve robustness in dense traffic scenes. For instance, in no-helmet detection, a transient occlusion caused by another rider or vehicle should not trigger an immediate false alarm if the decision is accumulated over a short observation window. Similarly, triple-riding decisions become more stable when rider counts are verified across consecutive frames rather than inferred from a single image.

ANPR performance will likely remain the most sensitive component of the system under night-time glare, motion blur, and plate occlusion. For this reason, the proposed architecture places ANPR after violation confirmation, thereby reducing unnecessary OCR computations and improving system efficiency.

2. Literature-Oriented Comparison

To position the proposed framework against representative prior studies, a feature-level comparison is useful. Earlier works commonly emphasize one or two tasks such as helmet

detection, plate localization, or vehicle-level ticketing. The proposed system differs by integrating these tasks into a unified evidence-generation pipeline.

Functional Coverage Comparison Between Prior Studies and Proposed System

Study	Focus	Helmet	Track	ANPR	Edge
Duong et al.	Helmet detection	✓	✓	✓	Limited
Charran & Dubey	Ticketing system	✓	✓	✓	Partial
Zhong & Liu	Safety + plate	✓	✓	✓	Limited
Chung et al.	Plate detection	×	Limited	✓	Limited
Proposed	Integrated system	✓	✓	✓	✓

Helmet = Helmet / rider reasoning; **Track** = Tracking; ANPR = ANPR / OCR, Multi-violation support, Edge = Edge deployment, **Multi** = Edge deployment.

The comparison highlights that the proposed contribution is not merely an improvement in one detector, but a broader system integration effort. This is important from an enforcement perspective because traffic agencies require end-to-end event generation rather than disjoint analytics modules.

3. Practical Strengths of the Proposed Approach

The proposed framework has several practical strengths:

1. It supports multiple violation classes within one deployable architecture.

2. It produces evidence packs, which gives the system practical enforcement value beyond raw detection.

3. It is suitable for edge deployment through model compression and acceleration.

4. It can be extended to additional rules such as mobile phone usage, overspeeding, and seat-belt violations.

5. It is designed around Indian traffic complexity, which improves its local relevance.

4. Expected Challenges

Certain challenges must be addressed during implementation and testing:

1. Heavy occlusion in dense traffic may reduce rider and helmet association accuracy.

2. Night-time scenes can degrade plate readability and increase false detections.

3. Red-light detection requires reliable signal-phase synchronization and stop-line calibration.

4. Wrong-lane detection depends on accurate lane geometry and camera perspective correction.

5. Legal admissibility may require careful timestamp integrity, data retention policy, and privacy-preserving redaction.

5. Privacy and Ethical Considerations

Because surveillance systems can affect public privacy, the proposed framework should incorporate selective evidence retention, access-controlled dashboards, and optional blurring of non-violator faces or unrelated vehicles. These design choices are important for aligning technical deployment with responsible AI and public-sector compliance expectations.

DISCUSSION AND CONCLUSION

This manuscript has presented a journal-aligned methodology for an AI-powered traffic violation detection and evidence generation system inspired by the project review presentation in the workspace. The work advances beyond isolated rule detection by proposing a unified enforcement pipeline that connects object detection, tracking, ANPR, OCR, and structured evidence creation. The resulting system is designed not only to identify violations but also to transform them into actionable digital records suitable for dashboard review and downstream enforcement workflows.

The strongest contribution of the proposed framework lies in its integration. Traffic surveillance systems often fail to transition from visual analytics to usable enforcement support because they stop at object detection. In contrast, the proposed approach explicitly models temporal consistency, contextual violation logic, and evidentiary output. This makes the system more relevant for real-world smart-city deployment.

At the same time, a careful limitation must be stated. The current manuscript captures the concept, architecture, and methodological foundation of the system, but it does not yet include field-tested numerical results. Future work should therefore focus on dataset collection under Indian road conditions, model training, camera calibration, threshold tuning, and controlled benchmarking on day-time and night-time traffic footage. Once these stages are completed, the manuscript can be extended with quantitative results, ablation studies, confusion matrices, and qualitative event samples from the deployed pipeline.

In conclusion, the proposed system represents a strong foundation for scalable and intelligent traffic law enforcement. With proper validation and deployment tuning, it can reduce manual effort, improve consistency in violation detection, and contribute to safer road ecosystems.

REFERENCES

*1. V. H. Duong, T. T. Nguyen, D. T. Pham, D. M. Nguyen, and S. W. Lee, "Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning," *IEEE Access*, vol. 8, pp. 162073-162082, 2020. DOI: [10.1109/ACCESS.2020.3021357](https://doi.org/10.1109/ACCESS.2020.3021357)*

*2. R. S. Charran and R. K. Dubey, "Two-Wheeler Vehicle Traffic Violations Detection and Automated Ticketing for Indian Road Scenario," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22002-22007, 2022. DOI: [10.1109/TITS.2022.3186679](https://doi.org/10.1109/TITS.2022.3186679)*

*3. S. Zhong and X. Liu, "A Safety Detection Method for Electric Bike Riding Incorporating License Plate Region Recognition," *IEEE Access*, vol. 13, pp. 124556-124568, 2025. DOI: [10.1109/ACCESS.2025.3588506](https://doi.org/10.1109/ACCESS.2025.3588506)*

*4. M. A. Chung, Y. J. Lin, and C. W. Lin, "YOLO-SLD: An Attention Mechanism-Improved YOLO for License Plate Detection," *IEEE Access*, 2024. DOI: [10.1109/ACCESS.2024.3419587](https://doi.org/10.1109/ACCESS.2024.3419587)*

*5. J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," *IEEE Access*, vol. 9, pp. 11203-11225, 2021. DOI: [10.1109/ACCESS.2020.3047929](https://doi.org/10.1109/ACCESS.2020.3047929)*

*6. L. Qiu et al., "Lightweight Deep Learning Models for Seat Belt Detection in Intelligent Vehicles," *Applied Sciences*, vol. 14, no. 2, 2024.*

7. Ultralytics, "Ultralytics YOLO Documentation and Repository," [Online]. Available: https://github.com/ultralytics/ultralytics

8. PaddlePaddle, "PaddleOCR Toolkit," [Online]. Available: https://github.com/PaddlePaddle/PaddleOCR

9. NVIDIA, "Jetson Xavier NX Developer Kit," [Online]. Available: https://developer.nvidia.com/embedded/jetson-xavier-nx

10. AI City Challenge, "Traffic Surveillance Benchmark and Dataset," [Online]. Available: https://www.aicitychallenge.org

