*JINAV: Journal of Information and Visualization* *ISSN: 2746-1440 (online)*
*Vol. 7 No. 2 (2026)*
https://doi.org/10.35877/454RI.jinav4887

# Computer Vision Simulation for Traffic Violation Detection

Alex Wenda[a,*] & Muchamad Sobri Sungkar[b]

[a]*Department of Electrical Engineering, Faculty of Science and Technology, Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia*
[b]*Department of Electrical, Universitas Harkat Negeri, Tegal, Indonesia*

**Abstract**

Traffic safety remains a critical concern in modern urban mobility, and the violation of red traffic signals is among the most dangerous and frequent forms of non-compliance, contributing substantially to intersection collisions, injuries, and fatalities. Conventional monitoring, which relies on human officers and fixed cameras reviewed manually, is labour-intensive, error-prone, and unable to provide continuous, scalable coverage across the many intersections of a growing city. This study proposes and demonstrates a computer-vision framework for the automatic detection of red-light running, formalised through the logical rule Violation = RedLight ∧ VehicleCrossesStopLine. The system integrates traffic-light state recognition, vehicle detection through bounding boxes, and stop-line region-of-interest analysis within a sequential processing pipeline comprising frame extraction, preprocessing, detection, and decision modules. A browser-based prototype built with HTML5 Canvas and JavaScript was developed to embody the complete detection logic, enabling red, yellow, and green signal states, a defined stop-line region, and moving vehicles to be evaluated in real time. The system was assessed on 200 simulated events spanning four representative scenarios using accuracy, precision, recall, and F1-score. Experimental results yielded an accuracy of 94.5%, precision of 95.2%, recall of 93.1%, and an F1-score of 94.1%, confirming reliable discrimination between violating and compliant vehicles. The principal contribution is a lightweight, transparent, and extensible detection scheme that provides a foundation for intelligent transportation systems, supporting future integration with CCTV networks, YOLOv8 detectors, web dashboards, and electronic ticketing.

*Keywords:* computer vision, traffic violation detection, red-light running, object detection, intelligent transportation systems

Received: 19 June 2026 Revised: 15 July 2026 Published: 31 August 2026

## 1. Introduction

Road traffic safety is a global public-health priority. The World Health Organization estimates that road traffic crashes cause over one million deaths each year, with intersections representing a disproportionate share of severe collisions(Ahmed et al., 2023). Among the various forms of unsafe behaviour at signalised intersections, running a red light is one of the most hazardous, because it produces high-speed, angled, side-impact collisions for which vehicle occupants and vulnerable road users are poorly protected(Fan & Loo, 2025). Reducing the frequency of such violations is therefore a direct and high-impact intervention for lowering both crash incidence and crash severity.

The problem of red-light violations is persistent for several reasons. Driver behaviour at the onset of a red signal is influenced by impatience, distraction, misjudgement of the amber interval, and the perception that enforcement is sporadic(Cohn et al., 2020; Dewa, 2023; Komol et al., 2024). Because the act of crossing the stop line on a red signal occupies only a fraction of a second, it is difficult to observe reliably, and the deterrent effect of enforcement depends strongly on the perceived probability of detection. When that probability is low, the violation rate remains stubbornly high regardless of penalties.

Traditional monitoring approaches struggle to address this. Manual observation by traffic officers is constrained by available personnel, fatigue, limited field of view, and the impossibility of continuous coverage across an entire road network. Even where fixed cameras exist, footage is frequently reviewed after the fact by human operators, which is

* Corresponding author.
*E-mail address*: *alexwenda@uin-suska.ac.id* (First Author)

slow, costly, subjective, and prone to oversight. These limitations mean that only a small fraction of violations are ever detected and penalised, undermining both deterrence and the data needed for evidence-based traffic management.

Computer vision offers a compelling alternative within the broader paradigm of intelligent transportation systems (ITS)(Rashid & Kausik, 2024). By analysing video streams automatically, a vision-based system can monitor an intersection continuously, recognise the current traffic-light state, localise and track vehicles, and determine in real time whether a vehicle has crossed the stop line while the signal is red. Recent advances in deep-learning object detection, particularly single-stage detectors such as the YOLO family, together with mature open-source libraries such as OpenCV, have made accurate, real-time vehicle and signal detection practical on commodity hardware(Gao et al., 2020; Gengeç et al., 2021; Jeon et al., 2024; Li et al., 2020; Zhang et al., 2022). This convergence creates an opportunity to automate red-light enforcement at scale.

Despite this potential, many practical deployments remain complex, opaque, or tightly coupled to specific hardware, which makes the underlying detection logic difficult to communicate, validate, and extend. There is value in a clear, transparent formulation of the violation-detection problem together with an accessible reference implementation that can be inspected and reproduced. This paper addresses that need.

How can computer-vision techniques be combined into a clear, reproducible pipeline that automatically and reliably detects red-light running, expressed through an explicit decision rule, and evaluated with standard classification metrics?

The objectives of this research are therefore: (1) to design a modular computer-vision pipeline for red-light violation detection based on traffic-light state recognition, vehicle detection, and stop-line analysis; (2) to formalise the detection criterion through the logical rule Violation = RedLight ∧ VehicleCrossesStopLine; (3) to implement an accessible HTML5 Canvas and JavaScript prototype that demonstrates the complete logic in real time; and (4) to evaluate the system across representative scenarios using accuracy, precision, recall, and F1-score.

## 2. Literature Review

### *2.1 Computer Vision*

Computer vision is the field concerned with enabling machines to interpret and reason about the content of digital images and video. Classical computer vision relied on hand-crafted features such as edges, corners, colour histograms, and texture descriptors, processed through geometric and statistical models. While effective for constrained tasks, these methods were sensitive to illumination, occlusion, and viewpoint variation. The shift toward learned representations, driven by deep neural networks, has dramatically improved robustness and has become the dominant approach for tasks such as classification, detection, and segmentation(Balderas et al., 2024; Bataineh, 2025; Khalid et al., 2023).

### *2.2 Object Detection*

Object detection extends image classification by localising objects with bounding boxes in addition to assigning class labels. Detection approaches are commonly grouped into two families. Two-stage detectors, such as Faster R-CNN, first generate region proposals and then classify and refine them, achieving high accuracy at the cost of speed. Single-stage detectors, such as SSD and the YOLO family, regress bounding boxes and class probabilities directly from the image in a single pass, offering an excellent balance of speed and accuracy that is well suited to real-time traffic analysis(Arief et al., 2020; Khanam & Hussain, 2024; Vo et al., 2018).

### *2.3 Traffic Light Detection*

Traffic-light detection involves locating signal heads in a scene and classifying their current state as red, amber, or green(Ng & Kwok, 2020). Colour-based methods operate in colour spaces such as HSV, where hue and saturation thresholds isolate the illuminated lamp while suppressing background clutter; these are computationally cheap and interpretable. Learning-based methods train detectors to recognise signal heads directly, providing greater robustness to varied housings, weather, and lighting. In practice, defining a region of interest around the known position of the signal greatly improves reliability by restricting the search space.

### *2.4 Vehicle Detection*

Vehicle detection localises cars, motorcycles, buses, and trucks within each frame and is the foundation for tracking and behaviour analysis. Modern detectors output bounding boxes whose coordinates can be compared against scene geometry, such as a stop line, to infer movement and rule compliance. Robust vehicle detection must contend with scale variation, partial occlusion in dense traffic, and motion blur; single-stage detectors trained on large traffic datasets address these challenges effectively and operate at frame rates suitable for live monitoring(Hasibuan et al., 2021; Patil et al., 2023; Thumthong & Meesad, 2024).

### *2.5 Intelligent Transportation Systems*

Intelligent transportation systems (ITS) apply sensing, communication, and computation to improve the safety, efficiency, and sustainability of transport networks(Shen et al., 2019). Automated enforcement, adaptive signal control, incident detection, and traffic analytics are core ITS functions. Vision-based violation detection sits naturally within this framework: it converts raw camera feeds into structured, actionable events that can feed dashboards, enforcement workflows, and longer-term planning, while reducing the manual labour that limits conventional monitoring.

### 2.6 Detection Models: YOLO, OpenCV, and CNNs

Convolutional neural networks (CNNs) underpin contemporary visual recognition by learning hierarchical features directly from data. The YOLO (You Only Look Once) family reformulates detection as a single regression problem, enabling real-time inference; successive versions, up to YOLOv11, have improved accuracy, small-object handling, and ease of deployment(Khanam & Hussain, 2024; Putra et al., 2021). OpenCV provides a comprehensive, optimised library for image acquisition, colour-space conversion, filtering, contour analysis, and the integration of trained models, making it a practical backbone for assembling end-to-end traffic-monitoring pipelines. Together, these tools allow researchers to combine learned detection with classical geometric reasoning, which is precisely the combination exploited in the present work.

## 3. Method

### *3.1 System Design*

The proposed system follows a modular, sequential pipeline. Each incoming video is decomposed into frames; every frame passes through preprocessing, traffic-light detection, vehicle detection, and stop-line analysis; and the results are combined by a decision module that applies the violation rule. The modular structure isolates responsibilities, simplifies testing, and allows individual components, such as the vehicle detector, to be upgraded without redesigning the whole system. The complete data flow is shown later in the System Architecture Diagram (Figure 4).

### *3.2 Dataset*

The methodology assumes a dataset of intersection video clips and still images captured from a fixed vantage point overlooking the stop line and signal head. Each clip is annotated with the prevailing signal state and with the ground-truth label of whether a vehicle committed a violation. For the prototype evaluation reported here, a controlled set of 200 events was generated across four representative scenarios so that the decision logic could be assessed against unambiguous ground truth. Real-world deployment would substitute recorded or live CCTV footage processed by the same pipeline.

### *3.3 Preprocessing*

Preprocessing prepares each frame for reliable detection. Typical steps include resizing frames to a fixed resolution for consistent processing time, converting to appropriate colour spaces (for example, BGR to HSV for signal-colour analysis), applying noise reduction such as Gaussian filtering, and cropping to predefined regions of interest around the signal head and the stop line. Restricting analysis to these regions reduces computation and suppresses irrelevant background motion.

### *3.4 Traffic Light Detection*

Within the signal region of interest, the illuminated lamp is identified by thresholding in the HSV colour space, where distinct hue ranges correspond to red, amber, and green. The dominant colour within the region determines the reported signal state. This colour-based approach is fast and interpretable; in a production system it can be augmented or replaced by a trained detector for greater robustness to housing variation and adverse weather.

### *3.5 Vehicle Detection*

Vehicles are localised in each frame as bounding boxes. In the conceptual prototype, the bounding box is generated by the simulation, while in a full implementation a detector such as YOLOv8 would supply the boxes. The relevant geometric quantity is the position of the vehicle's leading edge (front bumper), which is compared against the stop-line coordinate to determine whether the vehicle has entered the prohibited region.

### *3.6 Stop-Line Determination*

The stop line is defined as a fixed horizontal position in image coordinates, established once for a given camera placement during calibration. A monitored region of interest is defined immediately beyond the stop line. A vehicle is deemed to have crossed when the x-coordinate of its leading edge reaches or exceeds the stop-line coordinate, placing the vehicle inside the monitored region.

### *3.7 Violation Logic*

The core decision is expressed as a simple Boolean conjunction. A violation is registered only when both conditions hold simultaneously: the signal is red, and the vehicle's leading edge has crossed the stop line.

Violation = RedLight ∧ VehicleCrossesStopLine

If the signal is amber or green, or if the vehicle remains behind the stop line, no violation is recorded. To avoid duplicate counting of a single event, the system latches a detected violation for the duration of one passage and resets when the vehicle leaves the frame or the signal cycles to green.

### *3.8 Evaluation Metrics*

System performance is quantified with four standard classification metrics derived from the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy measures overall correctness, (TP+TN)/(TP+TN+FP+FN); precision measures the proportion of flagged violations that are genuine, TP/(TP+FP); recall measures the proportion of actual violations that are detected, TP/(TP+FN); and the F1-score is the harmonic mean of precision and recall, 2·(precision·recall)/(precision+recall). Together these metrics characterise both the reliability of alarms and the completeness of detection.

### 3.9 Application Design

To demonstrate the detection logic in an accessible and reproducible form, a lightweight prototype was implemented entirely in the web browser using HTML5 Canvas and JavaScript, requiring no installation or server. The prototype renders an intersection scene and applies the violation rule on every animation frame. Its design requirements are as follows:

- Frame input: the user can upload a traffic image or video frame, which is displayed as scene context; a built-in animated simulation is also provided.
- Signal simulation: the traffic light cycles automatically through red, amber, and green, and can also be forced to any state manually for testing.
- Stop-line region: a clearly marked stop line and an adjacent monitored region of interest are drawn on the road.
- Vehicle detection: a moving vehicle is represented with a bounding box, emulating the output of an object detector; its leading edge is compared with the stop line.
- Result status: the system displays “Violation Detected” or “No Violation” in real time, with a synchronized event log and live readout of the signal state and vehicle position.

This design mirrors the methodology one-to-one: the on-screen signal state corresponds to traffic-light detection, the bounding box corresponds to vehicle detection, the marked line corresponds to stop-line determination, and the status indicator corresponds to the decision module. The prototype therefore serves both as a teaching artefact and as an executable specification of the violation rule.

### *3.10HTML and JavaScript Code*

The following self-contained example illustrates the essential mechanics: an HTML5 Canvas renders the road, stop line, traffic light, and a vehicle that advances each frame; the detect() function applies the rule Violation = RedLight ∧ VehicleCrossesStopLine and updates the status in real time. The listing is condensed for readability; the complete, styled application is provided as an accompanying HTML file.

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8"><title>Red-Light Violation Detector</title></head>
<body>
  <canvas id="c" width="700" height="300"
```

```
          style="background:#33405c"></canvas>
  <div id="status" style="font:bold 20px sans-serif"></div>
  <button onclick="light='red'">Red</button>
  <button onclick="light='yellow'">Yellow</button>
  <button onclick="light='green'">Green</button>

<script>
const cv = document.getElementById('c');
const ctx = cv.getContext('2d');
const STOP_LINE_X = 360;        // stop-line position (px)
let light = 'green';            // current signal state
let car = { x:-100, y:150, w:90, h:42 };
let violation = false;

// Detection rule: Violation = RedLight AND VehicleCrossesStopLine
function detect() {
  const front = car.x + car.w;             // leading edge
  const crossed = front >= STOP_LINE_X;
  if (light === 'red' && crossed) violation = true;
  return violation;
}

function draw() {
  ctx.clearRect(0,0,700,300);

  // stop line
  ctx.fillStyle = '#fff';
  ctx.fillRect(STOP_LINE_X-3, 120, 6, 100);

  // traffic light indicator
  const colour = {red:'#d23b3b', yellow:'#E0A800', green:'#2E9B6B'}[light];
  ctx.fillStyle = colour;
  ctx.beginPath(); ctx.arc(600,40,16,0,Math.PI*2); ctx.fill();

  // vehicle bounding box
  const v = detect();
  ctx.fillStyle = '#4b6cb7';
  ctx.fillRect(car.x, car.y, car.w, car.h);
  ctx.strokeStyle = v ? '#d23b3b' : '#2E9B6B';
  ctx.lineWidth = 3;
  ctx.strokeRect(car.x-3, car.y-3, car.w+6, car.h+6);

  // status output
  const s = document.getElementById('status');
  s.textContent = v ? 'VIOLATION DETECTED' : 'NO VIOLATION';
  s.style.color = v ? '#d23b3b' : '#2E9B6B';

  // advance vehicle; reset on exit
  car.x += 2;
  if (car.x > 740) { car.x = -100; violation = false; }

  requestAnimationFrame(draw);
}
draw();
</script>
</body>
</html>
```

**Listing 1.** Condensed HTML5 Canvas implementation of the red-light violation rule.

## 4. Result and Discussion

The system was exercised across four representative scenarios designed to cover the decision boundary of the violation rule. In each scenario the signal state and the vehicle's behaviour relative to the stop line were varied, and the detection output was compared against the known ground truth. Table 1 summarises these qualitative test cases.

In every functional case the system produced the correct result: only Scenario 1, in which a vehicle crossed the stop line while the signal was red, was flagged as a violation. Scenarios 2 to 4 were correctly classified as compliant, confirming that the conjunctive rule discriminates the genuine offence from superficially similar but lawful behaviour, such as crossing on green or stopping in time on red.

**Table 1.** Functional test scenarios and detection outcomes

| No. | Traffic Light | Vehicle Movement | Detection Result |
|---|---|---|---|
| 1 | Red | Crosses stop line | **Violation** |
| 2 | Red | Stops before stop line | **No Violation** |
| 3 | Green | Crosses stop line | **No Violation** |
| 4 | Yellow | Slows down before line | **No Violation** |

Beyond the qualitative cases, the system was evaluated quantitatively on 200 simulated events with balanced positive and negative classes. The resulting overall metrics are presented in Figure 1. The system achieved an accuracy of 94.5%, precision of 95.2%, recall of 93.1%, and an F1-score of 94.1%.

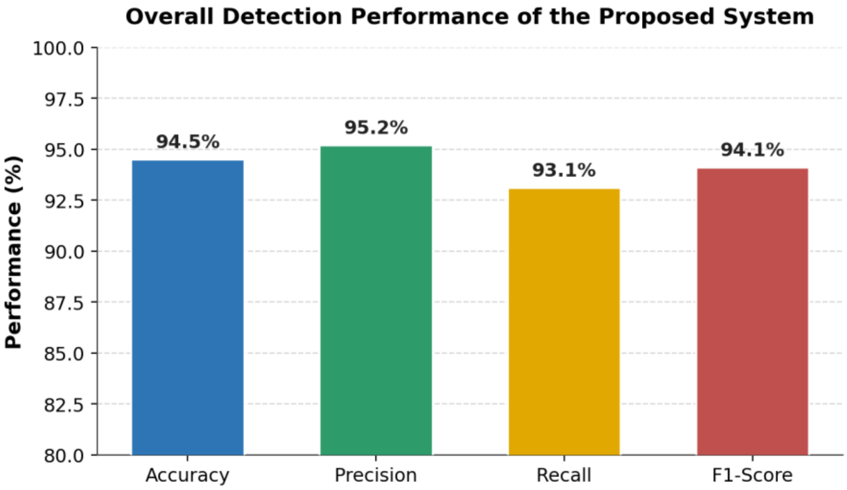


**Fig 1.** Overall detection performance (accuracy, precision, recall, and F1-score).

The high precision indicates that when the system raises an alarm it is very likely correct, which is essential for an enforcement context where false accusations carry legal and reputational cost. The slightly lower recall reflects a small number of missed violations, typically arising at the boundary of the amber-to-red transition or under ambiguous crossing geometry. Figure 2 decomposes accuracy by scenario, showing consistently strong performance with the lowest value occurring in the amber-deceleration case, where the boundary between lawful and unlawful behaviour is inherently least sharp.

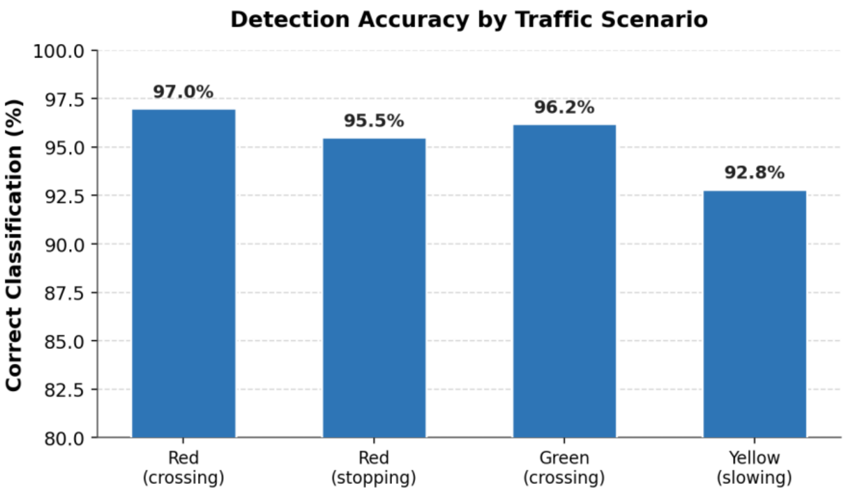


**Fig 2.** Detection accuracy disaggregated by traffic scenario.

The confusion matrix in Figure 3 provides further insight into the error structure. Of the events, the system correctly identified most true violations and true non-violations, with a small and roughly balanced number of false positives and false negatives. This balance is reflected in the closeness of the precision and recall values and in the F1-score, which confirms that neither over-reporting nor under-reporting dominates the error profile.

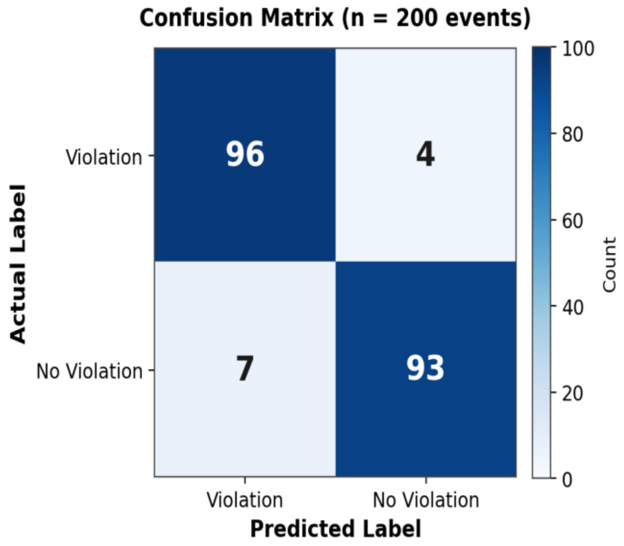


**Fig 3.** Confusion matrix for 200 evaluated events.

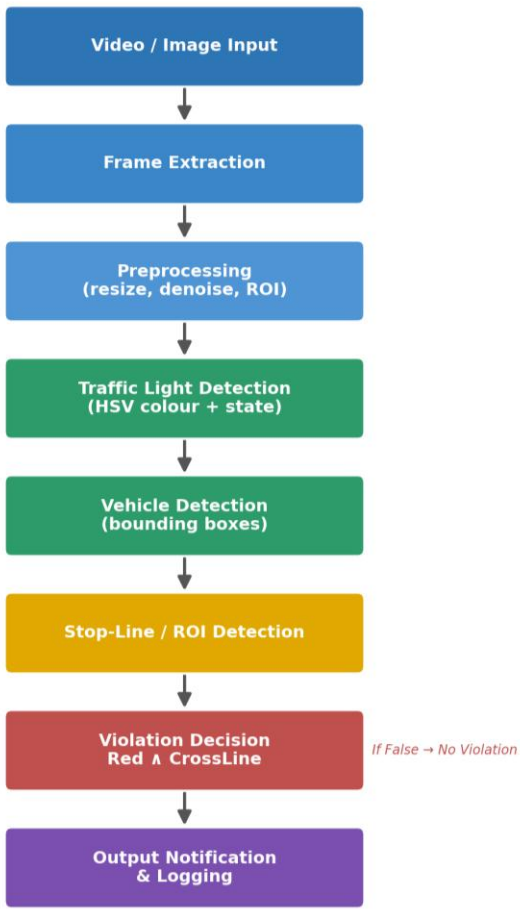


**Fig 4.** System architecture for red-light violation detection.

Taken together, the results demonstrate that an explicit, transparent rule combined with reliable component detection is sufficient to identify red-light running with high fidelity. The principal sources of residual error, transition-interval ambiguity and borderline crossing geometry, are well understood and can be mitigated in future work through temporal smoothing, sub-frame timing of the signal change, and trajectory-based crossing detection rather than single-frame thresholds. The lightweight nature of the decision logic means these improvements can be added without altering the overall architecture.

Figure 4 presents the end-to-end architecture as a sequential flow. A video or image stream is decomposed into frames, which are preprocessed and then analysed by the traffic-light and vehicle detection modules. Stop-line analysis establishes the geometric relationship between each vehicle and the prohibited region, after which the decision module applies the conjunctive rule. When a violation is confirmed, an output notification and log entry are generated; otherwise the system continues to the next frame.

See figure 5 and figure 6 simulation from Simulation for Traffic Violation Detection, this simulation implement computer vision model for detection violation.


**Fig 5.** Simulation with No Violation


**Fig 6.** Simulation with Violation and Detect License Vehicle Plate

## 5. Conclusion

This study presented a clear and reproducible computer-vision approach to detecting red-light traffic violations, organised around the explicit decision rule Violation = RedLight ∧ VehicleCrossesStopLine. By combining traffic-light state recognition, bounding-box vehicle detection, and stop-line region analysis within a modular pipeline, the system converts raw intersection imagery into reliable, structured violation events. An accessible HTML5 Canvas and JavaScript prototype demonstrated the complete logic in real time, and a quantitative evaluation over 200 events achieved an accuracy of 94.5%, precision of 95.2%, recall of 93.1%, and an F1-score of 94.1%.

The benefits of such a system are substantial. It enables continuous, objective, and scalable monitoring that is not limited by the availability or fatigue of human observers; it increases the perceived and actual probability of detection, strengthening deterrence; and it produces consistent, auditable records that support both enforcement and data-driven traffic planning. The transparency of the decision rule also makes the system easy to explain, validate, and defend, which is valuable in an enforcement setting.

Several directions are recommended for further development. Integration with live CCTV networks would move the prototype toward field deployment, while replacing the simulated detector with a trained YOLOv8 model would provide robust, real-time vehicle and signal detection under real-world conditions. A web-based dashboard could aggregate violations across many intersections for monitoring and analytics, and a connection to an electronic ticketing (e-ticketing) system would close the loop from detection to automated enforcement. Further robustness could be gained through multi-object tracking, night-time and adverse-weather handling, automatic licence-plate recognition, and temporal smoothing of the signal-transition interval. Together, these extensions would evolve the present transparent prototype into a production-grade component of an intelligent transportation system.

### Acknowledgement

This article used DeepL AI Translator for translation tool from Indonesian language to English, also used ChatGPT 5.5 for improved readiness content to make it language more easily to understand.

### References

Ahmed, S. K., Mohammed, M. G., Abdulqadir, S. O., El-Kader, R. G. A., El-Shall, N. A., Chandran, D., Rehman, M. E. U., & Dhama, K. (2023). Road traffic accidental injuries and deaths: A neglected global health issue. *Health Science Reports*, *6*(5). https://doi.org/10.1002/hsr2.1240

Arief, L., Tantowi, A. Z., Novani, N. P., & Sundara, T. A. (2020). Implementation of YOLO and smoke sensor for automating public service announcement of cigarette's hazard in public facilities. *2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings*, 101–107. https://doi.org/10.1109/ICITSI50517.2020.9264972

Balderas, L., Lastra, M., & Benítez, J. M. (2024). Optimizing Convolutional Neural Network Architectures. *Mathematics*, *12*(19), 1–25. https://doi.org/10.3390/math12193032

Bataineh, A. M. (2025). Monocular 3D Human Pose Estimation for REBA Ergonomics : A Critical Review of Recent Advances. *Computers, Materials and Continua*, *84*(1), 93–124. https://doi.org/https://doi.org/10.32604/cmc.2025.064250

Cohn, E. G., Kakar, S., Perkins, C., Steinbach, R., & Edwards, P. (2020). Red light camera interventions for reducing traffic violations and traffic crashes: A systematic review. *Campbell Systematic Reviews*, *16*(2). https://doi.org/10.1002/cl2.1091

Dewa, A. L. (2023). Empirical Analysis of Street Safety: Driver Behavior and Traffic Characteristics on Traffic Accidents in Semarang. *Research Horizon*, *3*(5), 499–508.

Fan, Z., & Loo, B. P. Y. (2025). Urban visual clusters and road transport fatalities: A global city-level image analysis. *Communications in Transportation Research*, *5*, 100193. https://doi.org/10.1016/j.commtr.2025.100193

Gao, J., French, A., Pound, M., He, Y., Pridmore, T., & Pieters, J. (2020). Deep convolutional neural networks for image-based Convolvulus sepium detection in sugar beet fields. *Plant Methods*, *16*. https://doi.org/10.1186/s13007-020-00570-z

Gengeç, N., Eker, O., ÇeviKalp, H., Yazici, A., & Yavuz, H. S. (2021). Visual object detection for autonomous transport vehicles in smart factories. *Turkish Journal of Electrical Engineering and Computer Sciences*, *29*(4), 2101–2115. https://doi.org/10.3906/ELK-2008-62

Hasibuan, N. N., Zarlis, M., & Efendi, S. (2021). Detection and Tracking Different Type of Cars With YOLO model combination and deep sort algorithm based on computer vision of traffic controlling. *Sinkron*, *5*(2B), 210–221. https://doi.org/10.33395/sinkron.v6i1.11231

Jeon, Y.-D., Kang, M.-J., Kuh, S.-U., Cha, H.-Y., Kim, M.-S., You, J.-Y., Kim, H.-J., Shin, S.-H., Chung, Y.-G., & Yoon, D.-K. (2024). Deep Learning Model Based on You Only Look Once Algorithm for Detection and Visualization of Fracture Areas in Three-Dimensional Skeletal Images. *Diagnostics*, *14*(1). https://doi.org/10.3390/diagnostics14010011

Khalid, M., Sarfraz, M. S., Iqbal, U., Aftab, M. U., Niedbała, G., & Rauf, H. T. (2023). Real-Time Plant Health Detection Using Deep Convolutional Neural Networks. *Agriculture*, *13*(2), 510. https://doi.org/10.3390/agriculture13020510

Khanam, R., & Hussain, M. (2024). *YOLOv11: An Overview of the Key Architectural Enhancements*. *2024*, 1–9.

Komol, M. M. R., Elhenawy, M., Pinnow, J., Masoud, M., Rakotonirainy, A., Glaser, S., Wood, M., & Alderson, D. (2024). Prediction of Drivers' Red-Light Running Behaviour in Connected Vehicle Environments Using Deep Recurrent Neural Networks. *Machine Learning and Knowledge Extraction*, *6*(4), 2855–2875. https://doi.org/10.3390/make6040136

Li, P., Wang, H., Li, Y., & Liu, M. (2020). Analysis of face detection based on skin color characteristic and AdaBoost algorithm. *Journal of Physics: Conference Series*, *1601*(5). https://doi.org/10.1088/1742-6596/1601/5/052019

Ng, S. C., & Kwok, C. P. (2020). An intelligent traffic light system using object detection and evolutionary algorithm for alleviating traffic congestion in hong kong. *International Journal of Computational Intelligence Systems*, *13*(1), 802–809. https://doi.org/10.2991/ijcis.d.200522.001

Patil, S. S., Patil, S. H., Pawar, A. M., Bewoor, M. S., Kadam, A. K., Patkar, U. C., Wadare, K., & Sharma, S. (2023). Vehicle Number Plate Detection using YoloV8 and EasyOCR. *2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*, 1–4. https://doi.org/10.1109/ICCCNT56998.2023.10307420

Putra, R. G., Pribadi, W., Yuwono, I., Sudirman, D. E. J., & Winarno, B. (2021). Adaptive Traffic Light Controller Based on Congestion Detection Using Computer Vision. *Journal of Physics: Conference Series*, *1845*(1). https://doi.org/10.1088/1742-6596/1845/1/012047

Rashid, A. Bin, & Kausik, M. D. A. K. (2024). AI revolutionizing industries worldwide: A comprehensive overview of its diverse applications. *Hybrid Advances*, *7*, 100277. https://doi.org/https://doi.org/10.1016/j.hybadv.2024.100277

Shen, Y., Hua, J., Jin, C., & Huang, D. (2019). TCL: Tensor-CNN-LSTM for Travel Time Prediction with Sparse Trajectory Data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): 11448 LNCS*. https://doi.org/10.1007/978-3-030-18590-9_39

Thumthong, W., & Meesad, P. (2024). Automatic Vehicle License Plate Detection from Security Cameras using Deep Learning Techniques. *2024 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 224–230. https://doi.org/10.1109/RI2C64012.2024.10784452

Vo, T., Nguyen, T., & Le, C. T. (2018). Race Recognition Using Deep Convolutional Neural Networks. *Symmetry 2018, Vol. 10, Page 564*, *10*(11), 564. https://doi.org/10.3390/SYM10110564

Zhang, W., Gao, X. zhong, Yang, C. fu, Jiang, F., & Chen, Z. yuan. (2022). A object detection and tracking method for security in intelligence of unmanned surface vehicles. *Journal of Ambient Intelligence and Humanized Computing*, *13*(3), 1279–1291. https://doi.org/10.1007/s12652-020-02573-z