**International Journal of Scientific Research in Engineering & Technology**
Volume6, Issue6 (November-December 2026), PP: 06-10
https://www.doi.org/10.59256/ijsreat.20260606002
www.ijsreat.com
ISSN No: 2583-1240

# AI-Based Smart Traffic Monitoring System

**Rekha BH[1], Kokila H D[2], Priyanka M S[3], Pruthvika H P[4], Abhishek M[5]**
[1]*Assistant Professor, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology/ Visvesvaraya Technological University, Karnataka, India.*
[2, 3, 4, 5]*Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology/ Visvesvaraya Technological University, Karnataka, India.*

***To Cite this Article:*** *Rekha BH[1], Kokila H D[2], Priyanka M S[3], Pruthvika H P[4], Abhishek M[5], "AI-Based Smart Traffic Monitoring System", International Journal Of Scientific Research In Engineering & Technology, Volume 06, Issue 06, November -December 2026, PP:06-10.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Abstract:** *Manual enforcement of traffic rules in Indian cities is slow, inconsistent and difficult to scale, and most automated systems treat every offence identically regardless of the offender's history. This paper presents the design and prototype implementation of the AI-Based Smart Traffic Monitoring System, built with Flask and OpenCV. The system registers five classes of camera-detectable violations (illegal parking, red-light jumping, riding without a helmet, wrong-way driving and triple riding) from simulated multi-camera feeds, uploaded images or video, and browser camera snapshots; stores evidence with each complaint; resolves owner details through a simulated RTO lookup; and uses a rule-based Violation Predictor to compute a Rule Breach Rate (15–98%) from prior violations and violation severity. The predictor then decides automatically between an official Warning for first-time offenders and a Direct E-Challan whose fine is escalated by 25% (third offence) and 50% (fourth and later offences), removing operator ambiguity at the review stage. A browser-side offline queue with idempotent retry prevents loss of evidence during server outages. Testing of the prototype showed that the decision logic behaves deterministically across all nine supported violation types, that image analysis takes 3–30 ms per 800×420 image on a CPU, and that the live scene renderer sustains the 45 ms stream interval comfortably. The detection layer is simulated and has not been validated on real CCTV data; this limitation, together with a weakness we found in the thumbnail-based demo matcher, is discussed along with the steps required for real deployment.*

**Keywords**: *Smart traffic management; automated enforcement; e-challan; repeat-offender prediction; OpenCV; Flask; ANPR; computer vision*

## I. INTRODUCTION

Road traffic violations such as illegal parking, riding without a helmet, triple riding and driving against the flow of traffic contribute directly to congestion and to road-safety risk in Indian cities. Enforcement is still largely manual: an officer must observe the offence, record the vehicle number, look up the owner and decide on a penalty. This approach does not scale to the number of junctions that need monitoring, and the penalty applied can vary between officers for identical offences.

Automated enforcement with cameras and computer vision addresses part of this problem. Real-time object detectors such as YOLO [1] and multi-object trackers such as ByteTrack [2] and DeepSORT [3] make vehicle detection and tracking practical, and automatic number-plate recognition (ANPR) can identify the registered vehicle [4]. However, a detection pipeline alone does not settle what action should follow. Under the Motor Vehicles (Amendment) Act, 2019 [5], penalties are defined per offence, and repeat offences are treated more seriously in practice. A system that treats a first-time and a habitual offender in the same way either over-penalises the former or under-penalises the latter.

Vision-based traffic monitoring generally follows a detect–track–classify pipeline. Single-stage detectors such as YOLO[1] provide the detection stage at real-time speed, while trackers such as SORT-family methods [2], [3] maintain vehicle identity across frames, which is necessary for violations defined by motion, such as red-light jumping and wrong-way driving. For plate reading, classical OCR engines [6] and detector-based pipelines designed for unconstrained scenes [4] have been used. Edge-based processing with the canny operator [7] remains a lightweight way of obtaining object contours and is available in OpenCV [8].

Most published systems concentrate on the accuracy of these perception stages. Comparatively little attention is given to the downstream enforcement decision and to operational robustness, such as what happens to evidence when the back-end is unreachable. The present prototype addresses these two aspects and substitutes simulated perception so that the workflow can be demonstrated without a GPU, labelled dataset or live CCTV access.

This work therefore focuses on the workflow that surrounds detection: evidence-first complaint registration, owner lookup, a transparent breach-rate computation, and an automated decision between Warning and E-Challan.

**Objectives:**

- To develop an end-to-end, laptop-runnable prototype that registers violations from simulated multi-camera feeds, uploaded media and browser snapshots, with evidence stored against every complaint.
- To implement a rule-based Violation Predictor that converts prior-violation count and violation severity into a Rule Breach Rate and an automated Warning/E-Challan decision, with fines escalated by 25% and 50% for repeat and habitual offenders.

- To provide a resilient registration mechanism in which the browser captures and queues evidence during a server outage and retries idempotently, so that no complaint is lost or duplicated.
- To give an honest assessment of the prototype, including measured behaviour of its components and the limitations that must be removed before real-world use.

## II. MATERIAL AND METHODS

### Study Design

Design and prototype implementation of a software system, followed by functional and performance testing of its components.

### Implementation Platform

Python with Flask (web server and REST endpoints), OpenCV and NumPy (image processing and scene rendering), and HTML/CSS/JavaScript templates for the officer interface. Complaint records are persisted in a JSON store. Optional components (Ultralytics YOLO, EasyOCR, Twilio, SendGrid) are listed as commented dependencies and are not required to run the prototype.

### System Architecture

The system is organised into the layers shown in Fig 1. Input is received from five simulated cameras, from image or video upload, or from a browser camera snapshot. The detection layer analyses the evidence and maps it to a violation type. A vehicle-lookup module returns the owner's record and prior violation count. The Violation Predictor combines these to decide the enforcement action. Complaints, with their evidence, are stored and surfaced on a dashboard and in a Control Room where an officer acts on pending complaints. A notifier then dispatches the result to the owner.

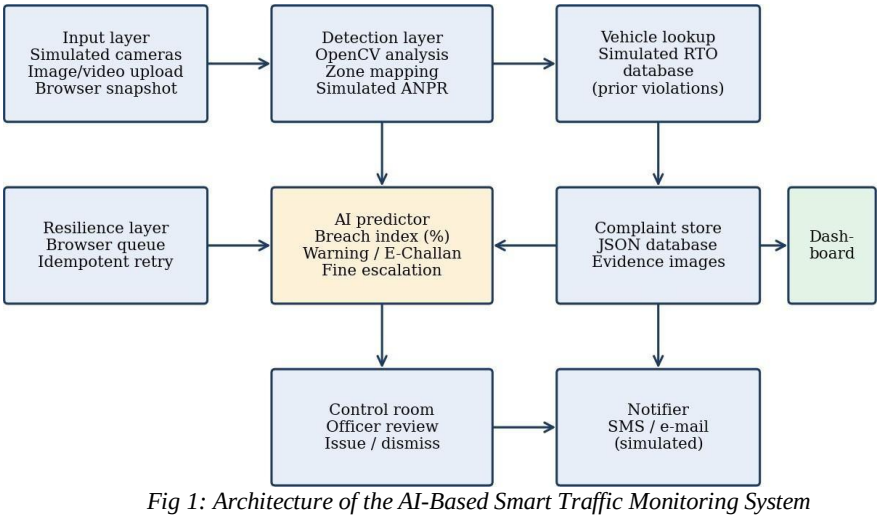



*Fig 1: Architecture of the AI-Based Smart Traffic Monitoring System*

### Camera Zones and Violation Types

Each camera is bound to a zone and a violation class (Table 1). The live feed is generated by an OpenCV scene renderer that draws animated vehicles, motorcycles, pedestrians, traffic signals, road markings, number plates, tracking boxes and a heads-up display. Violation events are generated within these scenes and can be registered as complaints. For uploaded evidence, the violation type is inferred from the zone name, and the image is analysed with Gaussian blur, Canny edge detection and contour extraction; contours with area above 2000 px² are counted as vehicle candidates, and the reported confidence is min(0.60 + 0.05 × contours, 0.99), or 0.55 when none are found. For video, the first twelve frames are sampled in the same way.

| Camera | Zone | Violation class |
|---|---|---|
| CAM-1 | West Shoulder – No Parking | Illegal parking (shoulder / restricted bay) |
| CAM-2 | Main Junction – Red Light | Red-light violation (stop-line crossing) |
| CAM-3 | Market Road – Helmet Check | Rider without helmet (two-wheeler safety) |
| CAM-4 | Ring Road – Wrong Way | Wrong-way driving (direction tracking) |
| CAM-5 | City Junction – Triple Riding | Triple riding (three persons on one bike) |

*Table 1: Simulated cameras and their violation zones*

### Simulated ANPR and Owner Lookup

Because the prototype has no trained plate reader, an upload or snapshot with no typed plate receives a plate suggestion from a deterministic simulated ANPR function. It derives a plausible Indian-format plate (state code, district, two letters, four digits) and a confidence between 0.62 and 0.92 from a SHA-256 hash of the image bytes. The result is labelled as a simulation and is editable by the officer before submission. The owner lookup likewise simulates an RTO/VAHAN query: a seeded random

generator keyed on the plate returns owner name, contact details, address, vehicle type, make, colour, registration validity and a previous-violation count between 0 and 5. Both modules are intended as replaceable interfaces for real ANPR and RTO integrations.

**Violation Predictor**

The Violation Predictor takes the plate, violation type, the RTO previous-violation count and the number of earlier complaints for that plate in the system. The prior count n is their sum. Each violation type has a severity weight s and a base fine (Table 2). The Rule Breach Rate (breach index) is computed by the piecewise rules in Table 3 and bounded within each band. The decision is Warning when n = 0 and Direct E-Challan when n ≥ 1, with the fine multiplied by 1.0, 1.25 and
1. for n = 1, n = 2 and n ≥ 3 respectively.

| Violation type | Severity weight s | Base fine (INR) | Priority |
|---|---|---|---|
| Illegal parking | 20 | 500 | Medium |
| Rider without helmet | 25 | 500 | Medium |
| Double parking | 35 | 1000 | Medium |
| Footpath encroachment | 35 | 750 | Medium |
| Triple riding | 50 | 1000 | High |
| Bus-stop obstruction | 60 | 2000 | Medium |
| Wrong-way driving | 70 | 1500 | High |
| Red-light violation | 75 | 1000 | High |
| Fire-lane violation | 85 | 3000 | High |

*Table 2: Severity weights and base fines (INR) for supported violation types*

| Prior n | Breach index (%) | Action | Fine |
|---|---|---|---|
| 0 | min(28, max(18, round(18 + s/10))) | Warning notice | 0 |
| 1 | min(68, max(55, round(55 + s/8))) | Direct E-Challan | Base fine |
| 2 | min(85, max(75, round(75 + s/10))) | E-Challan (repeat) | Base fine × 1.25 |
| ≥ 3 | min(98, round(85 + min(13, 3(n−2)) + s/25)) | E-Challan (habitual) | Base fine × 1.50 |

*Table 3: Decision rules of the Violation Predictor (n = prior violations)*

**Automated Direct Action and Notification**

Reviewing officers are not asked to choose between a warning and a fine. The Control Room shows the single action button that corresponds to the predictor's decision, together with a plain-language reason (for example, the number of prior breaches and the escalation applied). The officer may still dismiss a complaint. When an action is confirmed, the notifier composes an SMS and e-mail naming the vehicle, offence, zone, time, complaint reference, fine amount and breach rate, and logs the dispatch. Transport is simulated; Twilio and SendGrid are noted as production replacements.

**Resilient Evidence Capture**

Registration from the Register Violation and Live pages is submitted as JSON to a dedicated endpoint. The server writes the captured still image to a separate evidence folder and links it to the complaint. If the request fails, the browser captures the visible frame, stores the plate, zone, notes and image in local storage, shows a queued-offline banner, and retries on a timer and at the next page load. Each submission carries a client reference, so a retry whose first attempt actually succeeded does not create a second complaint. Recovered complaints are flagged and counted on the dashboard. Large video uploads use the standard multipart form because they are unsuitable for local storage.

**Security Measures nd Testing Procedure**

Officer sign-in requires an operator name and a password, which is read from an environment variable (a demo default is provided and must be changed before deployment). Uploads are restricted to PNG, JPEG, MP4, AVI and MOV extensions and checked against file signatures, and security headers are added to responses. Testing consisted of (i) enumerating the predictor output for every violation type and prior count; (ii) timing the image-analysis routine on the four bundled sample images and the scene renderer over 200 frames; (iii) checking determinism of the simulated ANPR; and (iv) measuring pairwise differences between the bundled images under the demo matcher, which downsamples to 32×32 grey scale and accepts a match when the mean absolute difference is below 6.0. Timing used a CPU-only development environment and should be read as indicative.

## III. RESULT

**Predictor Behaviour**

Table 4 lists the breach index produced for five representative violation types as the number of prior violations rises, and Fig 2 plots it. The index is monotonically non-decreasing inn, remains inside the intended 15–98% range, and for any given n is higher for more severe offences. Every first-time offence receives a warning, and every offence with n ≥ 1 receives an E-Challan.

| Violation type | n = 0 | n = 1 | n = 2 | n = 3 | n = 4 | n = 5 |
|---|---|---|---|---|---|---|
| No parking | 20 | 58 | 77 | 89 | 92 | 95 |
| No helmet | 20 | 58 | 78 | 89 | 92 | 95 |
| Red light | 26 | 64 | 82 | 91 | 94 | 97 |
| Wrong way | 25 | 64 | 82 | 91 | 94 | 97 |
| Fire lane | 26 | 66 | 84 | 91 | 94 | 97 |

*Table 4: Breach index (%) by number of prior violations*

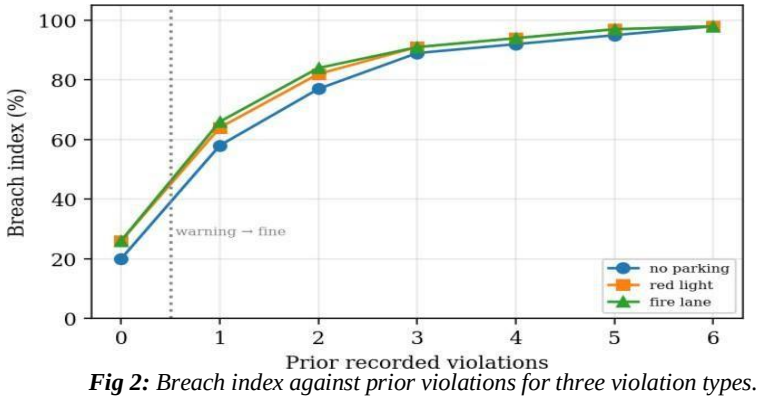



***Fig 2:*** *Breach index against prior violations for three violation types.*

| Violation type | n = 0 | n = 1 | n = 2 | n ≥ 3 |
|---|---|---|---|---|
| No parking | Warning | 500 | 625 | 750 |
| No helmet | Warning | 500 | 625 | 750 |
| Red light | Warning | 1000 | 1250 | 1500 |
| Wrong way | Warning | 1500 | 1875 | 2250 |
| Fire lane | Warning | 3000 | 3750 | 4500 |

*Table 5: Automated fine (INR) by number of prior violations (n = 0 yields a warning)*

Table 5 shows the fine escalation. For example, a wrong-way offence costs INR 1500 at n = 1, INR 1875 at n = 2 and INR 2250 from n = 3 onward, while a fire-lane offence rises from INR 3000 to INR 4500. The escalation is capped at +50%, so the penalty does not grow without limit for chronic offenders.

**Seeded Demonstration Cases**

On a fresh install the system seeds four complaints from the bundled evidence. Their automated outcomes (Table 6) show all three behaviours: a first-time helmet offence is warned, a repeat parking offence receives an escalated fine, and a habitual parking offender is assigned a 95% breach index.

| Plate | Violation | Breach index (%) | Decision | Status at seeding |
|---|---|---|---|---|
| KA01AB1234 | Illegal parking | 77 | E-Challan | Pending |
| KA03CD5678 | Red-light violation | 64 | E-Challan | Pending |
| KA05EF9012 | Rider without helmet | 20 | Warning | Pending |
| KA05MN7281 | Illegal parking | 95 | E-Challan | Pending |

*Table 6: Automated decisions for the four seeded complaints*

**Component Measurements**

| Measure | Result |
|---|---|
| Image analysis time, 800×420 image | 3.3–3.7 ms (30.3 ms on first call, including warm-up) |
| Contours detected per bundled image | 2 (reported confidence 0.70 for each) |
| Scene render time per frame (200 frames) | 1.25 ms mean; stream interval is 45 ms (about 22 frames/s) |
| Simulated ANPR determinism | Same image bytes gave the same plate and confidence |
| Simulated ANPR confidence range | 0.62–0.92 by construction |
| Pairwise difference between bundled images (32×32 grey, mean abs. diff.) | 0.04–0.09 grey levels, against a match threshold of 6.0 |

*Table 7: Measured behaviour of prototype components*

The measurements show that the pipeline is light enough for an ordinary laptop. They also expose a weakness: the four bundled sample images differ by less than 0.1 grey levels at the matcher's resolution, far below the threshold of 6.0, because they share the same background. When a test registry containing all four images was queried, each image matched the first registry entry rather than its own. The thumbnail matcher therefore cannot distinguish between the bundled scenarios, and the exact-violation behaviour described for uploaded copies of sample images depends on additional mapping logic rather than on this matcher alone.

## IV. DISCUSSION

The prototype shows that the enforcement decision can be separated cleanly from perception and made transparent. Because the Rule Breach Rate and the fine follow explicit rules, an officer or a citizen can see why a particular action was chosen, and the same offence history always produces the same outcome. This addresses the inconsistency of manual enforcement. Presenting a single action to the officer removes the need to choose between a warning and a fine, though the dismiss option keeps a human in the loop for disputed or erroneous detections, which we consider essential for any system that issues penalties.

The offline queue with idempotent retry is a practical contribution to robustness: evidence is captured at the moment of the event and survives server restarts, and retries cannot duplicate a complaint. Storing images in browser local storage is, however, limited by browser quotas, which is why video uses a separate path.

### Limitations

- Perception is simulated. Violations are generated by a scene renderer and by zone naming, and the system has not been evaluated on real CCTV footage. No detection accuracy, precision or recall is claimed. Real use requires a trained detector and tracker (for example YOLO with ByteTrack), camera calibration and a labelled dataset.
- ANPR and RTO data are simulated. Plates suggested from image hashes and owner records generated from a seeded random generator are not real data and must be replaced by plate OCR and an authorised VAHAN integration.
- The breach index is a rule-based heuristic, not a statistically trained model. Its bands and the 25%/50% escalation are design choices and have not been calibrated against real recidivism data or checked against the fine schedule of a particular state.
- The demo image matcher cannot discriminate between the bundled images (see Component Measurements) and should be replaced by perceptual hashing with a tighter threshold or by actual detection.
- The JSON store, shared demo password and simulated notifications are suitable for demonstration only. Deployment would need a database, role-based authentication, audit logging, a legal appeal workflow and data-protection safeguards for personal information.

### Future Work

Planned extensions are integration of a trained YOLO detector with tracking for motion-based violations, real plate recognition, a learned risk model trained on historical challan data, a payment and appeal workflow, and a field evaluation that reports detection accuracy and false-positive rates before any penalty is issued automatically.

## V. CONCLUSION

This paper presented the AI-Based Smart Traffic Monitoring System, which registers traffic violations with evidence, predicts a Rule Breach Rate from offence history and severity, and automatically chooses between a Warning and an escalating E-Challan. The prototype ran reliably on a CPU-only machine, behaved deterministically across all supported violation types, and preserved evidence through simulated server outages. Its detection, plate recognition and owner data are simulated, so it is a demonstration of the enforcement workflow and decision logic and not a validated enforcement tool. Replacing the simulated components with trained models and authorised data sources, and validating them on real footage, is the next step.

## REFERENCES

1. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, real-time object detection. Proc IEEE Conf Computer Vision and Pattern Recognition (CVPR). 2016:779–788.
2. Zhang Y, Sun P, Jiang Y, et al. ByteTrack: Multi-object tracking by associating every detection box. Proc European Conf Computer Vision (ECCV). 2022.
3. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. Proc IEEE Int Conf Image Processing (ICIP). 2017:3645–3649.
4. Silva SM, Jung CR. License plate detection and recognition in unconstrained scenarios. Proc European Conf Computer Vision (ECCV). 2018:580–596.
5. Government of India. The Motor Vehicles (Amendment) Act, 2019.
6. Smith R. An overview of the Tesseract OCR engine. Proc Int Conf Document Analysis and Recognition (ICDAR). 2007:629–633.
7. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986; 8(6):679–698.
8. Bradski G. The OpenCV Library. Dr. Dobb's Journal of Software Tools. 2000.