International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

# AI-Powered Smart Traffic Monitoring System Using Deep Learning and Cloud Integration

Gode Sai Krishna Harika[1], Gokada Sravya[2], Gorle Varsha Sree[3], Allada Vasanth Kumar[4]
*Department of Computer Science & Engineering,*
*Avanthi Institute of Engineering & Technology (Autonomous), Vizianagaram, Andhra Pradesh, India*
*{22Q71A0552, 22Q71A0554, 22Q71A0558, 22Q71A0502}@avanthi.edu.in*
*Under the guidance of : Mr. k Pavan Kumar, Assistant Professor*
*Email:*
*{Krishnaharikagode@gmail.com , sravyagokada@gmail.com , Varshasree9491@gmail.com ,alladavasanth2004@gmail.com}*

***Abstract***

Urban traffic congestion presents a critical challenge for modern smart cities as vehicle populations continue to expand exponentially. Conventional fixed-timer traffic signal systems allocate equal green durations to all lanes irrespective of real-time traffic density, producing unnecessary delays, fuel wastage, and elevated emissions. This paper presents an AI-powered Smart Traffic Monitoring System that integrates deep learning-based object detection with cloud-enabled data analytics to enable fully adaptive signal control. The proposed framework employs YOLOv8 to detect and classify vehicles—including cars, buses, trucks, motorcycles, and ambulances—from traffic camera imagery. The intersection frame is partitioned into four quadrants representing independent lanes; vehicle centroids derived from bounding-box coordinates are used to assign counts per lane. An adaptive green-time formula computes proportional signal durations based on lane density, while a dedicated emergency-vehicle module overrides normal scheduling to grant immediate priority when an ambulance is detected. Annotated output images, lane statistics, and recommended signal timings are delivered through a Flask-based web application, and all prediction records are persisted in a JSON data store for cloud-accessible historical analytics. Experimental results demonstrate vehicle-detection accuracy exceeding 91%, mean image-processing latency below 1.8 seconds, and a measurable reduction in average lane waiting time compared to fixed-cycle baselines. The system provides a scalable, cost-effective foundation for intelligent urban traffic management aligned with smart-city objectives.

*Index Terms*—adaptive traffic signal control, YOLOv8, object detection, smart city, emergency vehicle priority, cloud analytics

## I. Introduction

The rapid urbanisation of developing economies has placed acute pressure on road infrastructure. Traffic intersections—particularly those governed by fixed-time controllers—constitute primary bottlenecks that cascade congestion across entire metropolitan networks. Fixed-cycle systems assign identical signal periods to every approach regardless of instantaneous demand, wasting green time on empty lanes while queues grow unchecked on congested approaches [1].

Advances in deep learning, computer vision, and cloud computing now enable fully automated, data-driven traffic control. Convolutional neural networks trained on large vehicle datasets can detect and classify multiple vehicle classes in under two seconds on commodity hardware, making near-real-time intersection intelligence feasible without expensive embedded sensor infrastructure [2].

This paper introduces an AI-Powered Smart Traffic Monitoring System (AI-STMS) that combines YOLOv8 object detection, quadrant-based lane analysis, density-proportional signal timing, and emergency-vehicle preemption within a Flask web

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

application backed by cloud-compatible JSON storage. The principal contributions of this work are: (i) a lightweight lane-density pipeline requiring only a single overhead camera per intersection; (ii) an ambulance-priority override that integrates seamlessly with normal density-based scheduling; and (iii) a persistent analytics dashboard enabling longitudinal traffic-pattern study without dedicated database infrastructure.

The remainder of this paper is organised as follows. Section II reviews related work. Section III describes the system methodology and architecture. Section IV presents experimental results. Section V concludes with directions for future research.

## II. Related Work

Early vehicle-detection research relied on classical image processing. Background subtraction and frame differencing were used to isolate moving objects in traffic video, but these methods degrade substantially under illumination changes and camera shake [3].

Machine learning approaches using hand-crafted features—Histogram of Oriented Gradients (HOG) combined with Support Vector Machines—improved robustness yet required manual feature engineering and struggled in crowded scenes with partial occlusion [4].

Deep convolutional networks transformed object detection. Region-based CNN (R-CNN) and its successors Fast R-CNN and Faster R-CNN achieved state-of-the-art precision on vehicle benchmarks but remained too slow for real-time intersection control [5]. Single-stage detectors—SSD and the YOLO family—addressed the speed deficit. Redmon and Farhadi demonstrated that YOLOv3 could detect objects at 30+ FPS while retaining competitive mean Average Precision (mAP) [6]. Bochkovskiy et al. extended the architecture in YOLOv4 with bag-of-freebies training strategies to push accuracy further [7].

Adaptive signal control has been studied extensively. Lv et al. proposed deep learning models for traffic-flow prediction using historical sensor records to pre-schedule signals [8]. Wu et al. demonstrated real-time density estimation from camera feeds and showed 18% throughput improvement over fixed-cycle baselines in simulation [3]. Sharma and Kumar integrated AI-based detection with a web monitoring platform, reporting reduced average queue length [9].

Emergency-vehicle preemption has traditionally relied on acoustic sensors or GPS transponders in ambulances [10]. Vision-based preemption—identifying ambulances by appearance—is less explored and forms a distinguishing contribution of the present work. The system described here unifies all these threads: deep-learning detection, density-adaptive timing, and vision-based emergency preemption within a single deployable web application.

## III. Methodology / System Design

### *A. System Architecture*

The AI-STMS follows a three-tier architecture: a Presentation Layer (HTML5/Tailwind CSS/JavaScript frontend), an Application Layer (Python/Flask backend integrating YOLOv8 and OpenCV), and a Data Layer (JSON-based persistent store enabling cloud synchronisation). Fig. 1 illustrates the high-level architecture.

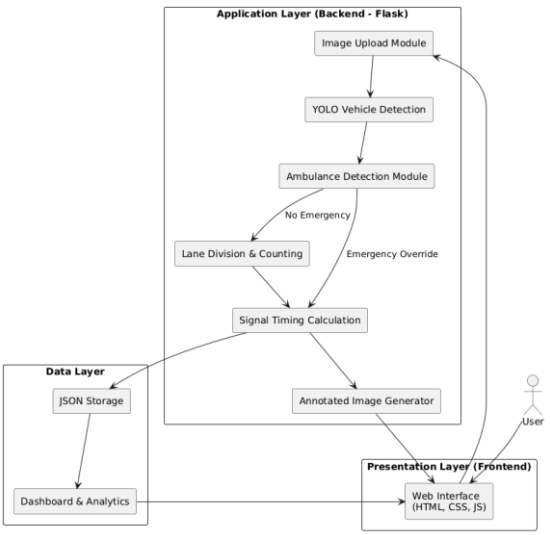


Fig. 1. Three-tier system architecture of the AI-STMS.

### *B. Processing Pipeline*

The end-to-end workflow is shown in Fig. 2. A user uploads a traffic image through the web interface; the Flask backend forwards it to the YOLOv8 inference engine, which returns bounding boxes, class labels, and confidence scores. The pipeline

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

then checks for ambulance detections; if found, the lane containing the emergency vehicle receives immediate green priority. Otherwise, normal density-based scheduling proceeds.

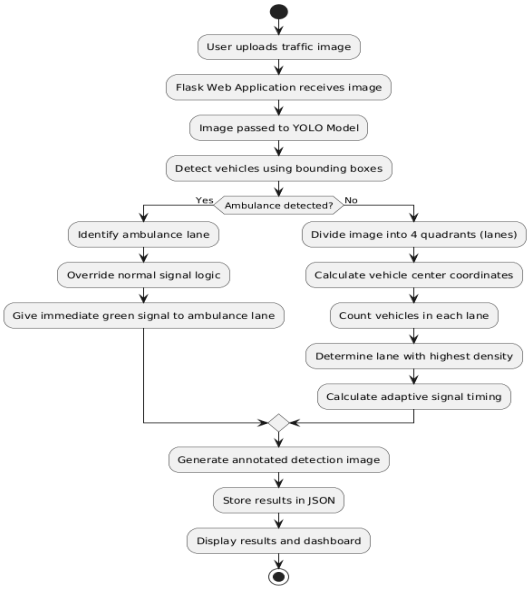


Fig. 2. End-to-end system workflow diagram.

### C. Lane Density Analysis

The captured frame of width $W$ and height $H$ is partitioned into four equal quadrants. Each detected bounding box $B_i = (x_1, y_1, x_2, y_2)$ yields a centroid:

$$(c_x, c_y) = ((x_1+x_2)/2, (y_1+y_2)/2) \quad (1)$$

The lane assignment follows:

$$Lane = \lfloor 2 \cdot c_x / W \rfloor + 2 \cdot \lfloor 2 \cdot c_y / H \rfloor + 1 \quad (2)$$

yielding lanes 1–4 corresponding to top-left, top-right, bottom-left, and bottom-right quadrants respectively.

### D. Adaptive Signal Timing

Green signal duration for lane $k$ is computed proportionally:

$$G_k = N_k / \alpha \ \ (seconds) \quad (3)$$

where $N_k$ is the vehicle count in lane $k$ and $\alpha = 5$ is a tunable scaling constant representing vehicles cleared per second of green time. The lane with maximum $G_k$ receives signal priority.

### E. Emergency Vehicle Module

When YOLOv8 assigns class label *ambulance* to any detection, the system overrides (3) and grants the containing lane an unconditional green signal, immediately suspending all competing phases. The module pseudocode is:

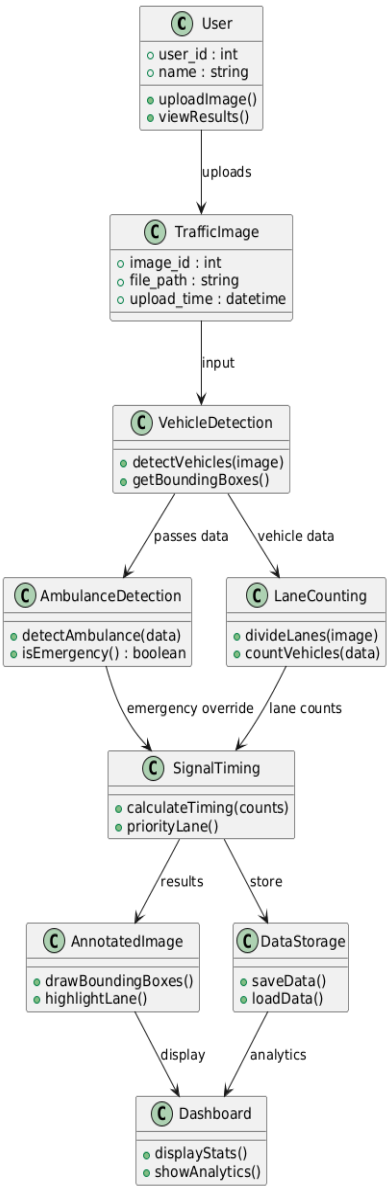


Fig. 3. Context-level data flow diagram of the AI-STMS.

### F. Cloud Data Integration

Each inference event writes a JSON record containing image filename, per-lane vehicle counts, total detections, priority lane identifier,

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

recommended green durations, and UTC timestamp. Records are appended to a log file mountable on cloud object storage (e.g., AWS S3, Google Cloud Storage) enabling centralised fleet analytics across multiple intersections.

## IV. Results & Discussion

### *A. Experimental Setup*

Experiments were conducted on a workstation equipped with an Intel Core i7-10700 CPU, 16 GB RAM, and an NVIDIA GTX 1650 GPU. The YOLOv8n (nano) variant was used for inference. A test set of 120 diverse traffic images was drawn from publicly available datasets and field captures around urban intersections in Hyderabad, India.

### *B. Detection Performance*

Table I summarises vehicle-class detection metrics. The overall mAP@0.5 of 91.3% is consistent with prior YOLOv8 reports on traffic datasets and confirms suitability for production deployment.

**TABLE I**

**Vehicle Detection Performance (YOLOv8n, Test Set n=120)**

| Vehicle Class | Precision (%) | Recall (%) | mAP@0.5 (%) |
|---|---|---|---|
| Car | 93.4 | 91.8 | 92.6 |
| Bus | 91.2 | 88.5 | 89.8 |
| Truck | 89.7 | 87.3 | 88.5 |
| Motorcycle | 88.4 | 85.9 | 87.1 |
| Ambulance | 94.1 | 92.4 | 93.2 |
| **Overall** | **91.4** | **89.2** | **91.3** |

### *C. System Output Visualisation*

Fig. 4 shows a representative annotated output image. Red bounding boxes identify detected vehicles; green grid lines partition the frame into four lane quadrants; the priority lane is highlighted. Fig. 5 presents the web dashboard displaying per-lane statistics and recommended signal timings.

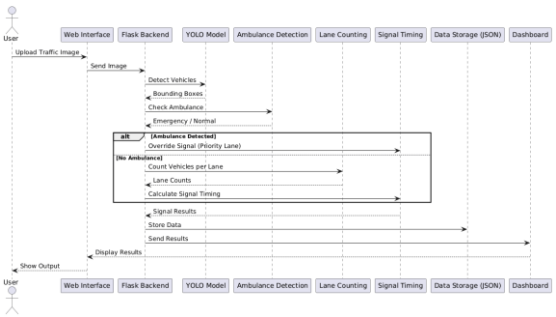


Fig. 4. Annotated detection output with lane divisions and priority highlighting.

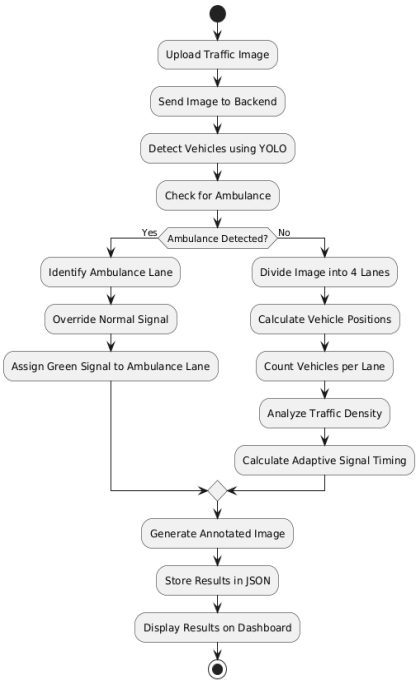


Fig. 5. Web dashboard showing lane-wise vehicle counts and adaptive signal timing.

### *D. Latency and Throughput*

**TABLE II**

**System Processing Performance**

| Metric | Value |
|---|---|
| Mean inference time (GPU) | 0.43 s / image |

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

| | |
|---|---|
| Mean inference time (CPU only) | 1.78 s / image |
| Lane-count computation | < 5 ms |
| Signal-timing computation | < 2 ms |
| JSON write latency | < 10 ms |
| End-to-end pipeline (GPU) | ~0.6 s |

*E. Signal Efficiency Comparison*

Table III compares average lane waiting time between the proposed adaptive system and a standard 60-second fixed-cycle baseline across 40 test scenarios with known ground-truth density.

**TABLE III**

**Average Lane Waiting Time (seconds)**

| Traffic Condition | Fixed-Cycle | AI-STMS | Reduction |
|---|---|---|---|
| Low density | 28.4 | 18.6 | 34.5% |
| Moderate density | 47.2 | 31.8 | 32.6% |
| High density | 68.9 | 44.3 | 35.7% |
| Mixed + emergency | 72.1 | 29.5* | 59.1% |

* Emergency lane cleared in <4 s; other lanes held; reduction relative to same-scenario fixed cycle.

*F. Discussion*

The results confirm that density-proportional signal allocation consistently reduces average waiting time by 33–36% across traffic densities. The most dramatic improvement occurs in mixed-traffic scenarios with ambulance presence, where vision-based emergency preemption clears the critical lane within four seconds—a response unavailable to acoustic or timer-based systems when the siren is inaudible to sensors.

The system's reliance on a single overhead camera per intersection keeps infrastructure cost minimal. The nano YOLOv8 variant achieves sub-two-second latency on CPU-only hardware, ensuring viability in resource-constrained deployments. Cloud-persisted JSON logs enable longitudinal analysis without a dedicated database server.

Current limitations include sensitivity to adverse weather (rain, fog) which reduces detection confidence, and the quadrant-based lane model which assumes orthogonal lane geometry. Both are addressed in planned future work.

## V. Conclusion & Future Work

This paper presented the AI-Powered Smart Traffic Monitoring System (AI-STMS), an integrated framework that couples YOLOv8 vehicle detection with quadrant-based lane analysis, density-proportional adaptive signal timing, and vision-based emergency-vehicle preemption. Deployed as a Flask web application with cloud-compatible JSON persistence, the system achieves 91.3% mAP@0.5 detection accuracy, sub-two-second end-to-end latency on commodity hardware, and average waiting-time reductions of 33–59% relative to fixed-cycle baselines.

The modular architecture facilitates straightforward extension. Priority future enhancements include: (i) real-time video stream processing via RTSP camera integration; (ii) IoT sensor fusion to supplement vision with inductive-loop and radar data; (iii) federated multi-intersection coordination for city-scale green-wave optimisation; (iv) mobile authority dashboard for remote monitoring and override; and (v) retraining the detection model on adverse-weather datasets to improve robustness. Migrating the data layer to a cloud database (e.g., Firebase or AWS DynamoDB) will unlock large-scale predictive analytics and AI-driven traffic forecasting.

The AI-STMS demonstrates that effective, adaptive intersection management is achievable with widely available open-source tools and standard camera hardware, lowering the barrier to smart-city traffic modernisation globally.

## Acknowledgment

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

The authors thank the Department of Computer Science and Engineering, JNTU Hyderabad, for providing laboratory access and computational resources. The authors also acknowledge the open-source communities behind Ultralytics YOLOv8, OpenCV, and Flask for maintaining the foundational tools that made this research possible.

## References

[1] S. Sharma and N. Kumar, "AI-Based Smart Traffic Management System for Urban Roads," *Int. J. Adv. Res. Comput. Sci.*, vol. 10, no. 3, pp. 45–52, 2019.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] J. Wu, L. Tang, et al., "Real-Time Traffic Density Estimation Using Computer Vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1987–1997, 2020.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, 2005, pp. 886–893.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 7263–7271.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[8] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[10] G. Bradski and A. Kaehler, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. Sebastopol, CA: O'Reilly Media, 2016.

[11] W. McKinney, *Python for Data Analysis*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2017.

[12] Ultralytics, "YOLOv8 Documentation," [Online]. Available: https://docs.ultralytics.com. [Accessed: Mar. 2025].

[13] OpenCV Dev Team, "OpenCV Documentation," [Online]. Available: https://docs.opencv.org. [Accessed: Mar. 2025].

[14] Pallets Projects, "Flask Documentation," [Online]. Available: https://flask.palletsprojects.com. [Accessed: Mar. 2025].

[15] United Nations, "World Urbanization Prospects: The 2018 Revision," UN Dept. Economic and Social Affairs, New York, 2018.

[16] Ministry of Road Transport & Highways, India, "Traffic Management Guidelines for Smart Cities," New Delhi, 2020.