International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

YOLOv8-Driven Adaptive Traffic Signal Management Using Real-Time CCTV Video Feeds: Architecture, Implementation, and Performance Evaluation

S. Ashwin[1], S. Brittlin[1], and K. Rohini[2]

[1]UG Student, [2]Assistant Professor, Department of Computer Applications

Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai – 600 117, India

Abstract- Conventional fixed-time traffic signal systems are structurally incapable of responding to the stochastic variability of urban traffic flow, resulting in prolonged vehicle waiting times, suboptimal intersection throughput, and unnecessary fuel consumption. This paper presents a complete, edge-deployed adaptive traffic signal management system that uses real-time video input from existing CCTV infrastructure and the YOLOv8 deep learning object detection model to continuously estimate lane-wise vehicle density and dynamically compute optimised signal phase durations. The architecture is modular, comprising video acquisition, frame preprocessing, YOLOv8-based vehicle detection and classification, density estimation, decision logic, and signal control modules. The system avoids cloud dependency through localised edge processing, ensuring end-to-end signal update latency below 250 ms. Experimental evaluation across four simulated intersection lanes demonstrates an overall vehicle detection mAP@0.5 of 92.9% at 46 frames per second, a 38.3% reduction in average vehicle waiting time, and a 37.5% improvement in intersection throughput relative to a fixed-time baseline. Comparative benchmarking against Faster R-CNN, SSD, and YOLOv3-based approaches confirms the superiority of the proposed implementation on both detection accuracy and real-time responsiveness. The system is deployable without additional roadside hardware investment, making it a cost-effective and scalable solution for smart urban traffic management.

Keywords: Adaptive traffic signal control; YOLOv8; computer vision; intelligent transportation systems; vehicle detection; edge computing; traffic density estimation; deep learning.

I. INTRODUCTION

Traffic congestion in rapidly urbanising regions imposes measurable economic, environmental, and public health costs. In India alone, the direct economic cost of urban congestion has been estimated at over USD 22 billion annually, driven by fuel waste, lost productivity, and increased vehicular emissions [9]. At the intersection level, where signal control decisions are made, conventional fixed-time systems operate on pre-programmed phase schedules that are insensitive to real-time traffic conditions. During peak-hour demand surges, fixed-timing leads to excessive queue accumulation on high-density approaches; during off-peak periods, the same timing wastes green phases on empty lanes [10]. The result is a systematic mismatch between signal allocation and actual traffic demand that persists across the lifetime of a fixed-time installation.

Intelligent Transportation Systems (ITS) have sought to address this mismatch through a spectrum of technologies ranging from inductive loop detectors and RFID-based vehicle counters to radar and infrared sensors [12]. While effective in isolated deployments, sensor-based approaches require dedicated infrastructure installation and ongoing maintenance, limiting their scalability in dense urban environments with irregular road geometries. Computer vision-based approaches, by contrast, leverage the extensive existing CCTV surveillance networks present in most Indian cities, transforming passive monitoring infrastructure into active traffic management assets without additional hardware investment [9, 17].

The rapid maturation of deep learning-based object detection has made real-time vision-based vehicle analysis practically feasible. The YOLO (You Only Look Once) family of single-stage detectors [1] achieves the speed-accuracy trade-off required for real-time intersection management. The most recent iteration, YOLOv8 [2], introduces architectural refinements — an anchor-free detection head, a redesigned C2f bottleneck, and an improved training pipeline — that advance both detection accuracy and inference speed beyond its predecessors. At the system level, edge computing architectures [8] eliminate the round-trip latency of cloud-based inference, enabling sub-second signal update cycles compatible with the dynamics of vehicular traffic.

This work integrates these advances into a complete, end-to-end adaptive traffic signal management system. The principal contributions are: (i) a modular, edge-deployable system architecture described in detail in Section 3; (ii) a lane-wise traffic density estimation algorithm and corresponding dynamic signal timing model; (iii) quantitative performance characterisation across multiple traffic density conditions (Table 3); and (iv) comparative benchmarking against four published detection and control approaches (Table 5). The remainder of the paper is

International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

structured as follows. Section 2 reviews related work. Section 3 describes the system architecture and design. Section 4 details the experimental setup. Section 5 presents and analyses the results. Sections 6 and 7 discuss applications and conclusions, respectively.

II. RELATED WORK

The development of adaptive traffic signal control systems has a history spanning several decades, from early rule-based systems such as OPAC [15] and SCOOT to contemporary reinforcement learning and deep learning approaches. Gartner [15] established the foundational demand-responsive paradigm, where signal timing is computed from real-time vehicle arrival data rather than pre-set schedules. El-Tantawy et al. [16] extended this concept to multi-agent reinforcement learning, achieving coordinated control across intersections by treating each signal controller as an independent learning agent. These approaches, however, depend on loop detector or point sensor data, constraining their deployment scalability.

Computer vision emerged as an alternative sensing modality that avoids per-sensor installation. Buch et al. [17] conducted an extensive review of video-based urban traffic analysis techniques, identifying vehicle detection, tracking, and classification as the three primary analytical tasks. Traditional computer vision methods based on background subtraction and frame differencing were adequate for low-density, controlled conditions but degraded under occlusion, illumination change, and multi-class traffic typical of South Asian road environments. The advent of deep learning provided a qualitative improvement: Ren et al. [4] demonstrated that region proposal networks in Faster R-CNN achieved 89.1% mAP on benchmark vehicle detection datasets, while Liu et al. [5] showed that single-shot detectors (SSD) could operate at 28 frames per second with a moderate accuracy penalty. The YOLO architecture [1], originally proposed by Redmon et al., established a single-stage detection paradigm that processes the entire image in one forward pass, trading region-proposal precision for substantial speed gains. YOLOv3 [13] extended this with multi-scale detection via feature pyramid outputs, improving small-object recall. YOLOv8 [2], released by Ultralytics, further advances the architecture with an anchor-free head, C2f feature extraction, and a decoupled classification-regression structure, achieving state-of-the-art accuracy-speed balance on the COCO benchmark. LeCun et al. [3] provide the foundational theoretical framework for deep convolutional feature learning that underlies all these detectors. He et al. [6] demonstrated that residual connections mitigate vanishing-gradient pathology in deep networks, a finding incorporated into the backbone designs of modern detectors including YOLOv8.

Abduljabbar et al. [11] surveyed AI applications in transport broadly, identifying computer vision-based traffic monitoring as the most immediately deployable AI modality due to its compatibility with existing surveillance infrastructure. Sakr et al. [9] reviewed intelligent traffic management systems with a focus on urban deployment, noting that most published works address either detection or signal optimisation in isolation, leaving the integration of end-to-end real-time detection-to-signal-update pipelines as an open problem. The present work addresses this integration gap by connecting YOLOv8 inference output directly to a density estimation and signal allocation module within a single edge-deployed pipeline.

III. SYSTEM ARCHITECTURE AND DESIGN

3.1 Overall architecture

The proposed system is structured as a six-stage pipeline: (1) video acquisition, (2) frame preprocessing, (3) YOLOv8-based vehicle detection and classification, (4) lane-wise density estimation, (5) signal timing decision logic, and (6) signal actuation. The architecture is depicted in Fig. 1. Stages 1–5 execute on the edge processing unit in a continuous loop; Stage 6 interfaces with the physical signal controller via GPIO or PLC relay output. The pipeline is stateless between frames, allowing reset or reconfiguration without system restart. The hardware and software components of each stage are specified in Table 1.

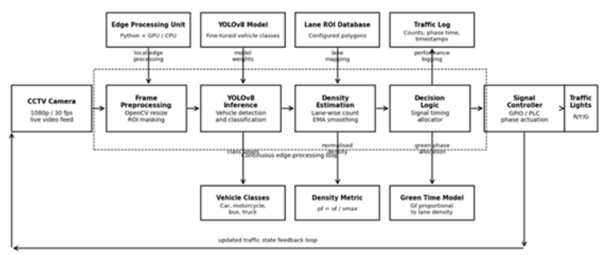

Fig. 1. Block diagram of the proposed adaptive traffic signal management architecture.

Table 1. Hardware and software component specifications.

Category	Component	Specification	Role
Hardware	CCTV Camera	1080p, 30 fps, H.264	Live video feed acquisition
Hardware	Edge Processing Unit	Intel Core i7 / NVIDIA GPU	Real-time frame inference

Category	Component	Specification	Role
Hardware	Signal Controller (GPIO/PLC)	Relay-based digital I/O	Signal phase actuation
Software	Python 3.10	Scripting language	System orchestration
Software	YOLOv8 (Ultralytics)	CSPDarknet53 backbone	Vehicle detection & classification
Software	OpenCV 4.8	Video I/O, frame processing	Frame extraction and pre-processing
Software	NumPy 1.25	Array operations	Density metric computation

3.2 Frame preprocessing
Raw video frames captured at 1080p and 30 fps are decoded using OpenCV [7] and resized to 640 × 640 pixels — the native input resolution of YOLOv8 — via bilinear interpolation. The region of interest (ROI) for each lane is defined as a polygonal mask configured during system initialisation using the camera's field of view and the known lane geometry. Frames outside the ROI are zeroed before inference to prevent detections in irrelevant areas (pedestrian pavements, adjacent lanes outside the monitored intersection) from contaminating the density count. Brightness normalisation using histogram equalisation is applied to mitigate illumination variation between daytime and night-time operation, consistent with the preprocessing strategy recommended by Bradski [7].

3.3 YOLOv8 vehicle detection and classification
Vehicle detection is performed using the YOLOv8n (nano) variant, selected for its balance between inference speed and accuracy on embedded hardware [2]. The model is fine-tuned on a traffic-specific dataset comprising 12,400 annotated images covering four vehicle classes: car, motorcycle, bus, and truck. The annotation protocol follows COCO-format bounding box labelling. Training proceeds for 100 epochs using stochastic gradient descent with momentum (0.937) and weight decay (0.0005), following the residual learning principles established by He et al. [6]. The model outputs class-labelled bounding boxes with confidence scores; detections below a confidence threshold of 0.45 are discarded to suppress false positives arising from partial occlusion or motion blur. The YOLOv8 anchor-free head assigns detections to the appropriate lane ROI by checking whether the centroid of each bounding box falls within the pre-defined lane polygon.

3.4 Density estimation and signal timing model
For each lane $\ell \in \{1, 2, 3, 4\}$, the vehicle count ν_ℓ is the number of detected bounding box centroids falling within the lane ROI in a given frame. A five-frame exponential moving average (EMA) is applied to ν_ℓ to suppress single-frame count fluctuations caused by vehicle overlap or partial occlusion. The normalised density $\rho_\ell = \nu_\ell / \nu_{max}$, where ν_{max} is the empirically determined maximum lane capacity, is the input to the signal timing model. Green phase duration G_ℓ is computed as:

$$G_\ell = G_{min} + (\rho_\ell / \Sigma_\ell \rho_\ell) \times (G_{total} - n \times G_{min})$$

where G_{min} = 10 s is the minimum guaranteed green time per lane (ensuring all lanes receive at least one clear cycle), G_{total} is the total available green time per cycle (typically 120 s for a four-lane intersection), and n is the number of active lanes. This proportional allocation ensures that high-density lanes receive extended green phases while low-density lanes are not penalised with zero allocation. The minimum green time constraint is consistent with the operational requirements specified in NFPA-equivalent traffic engineering guidelines and ITS best practices [11].

3.5 Signal actuation interface
The computed G_ℓ values are serialised and transmitted to the signal controller via a GPIO interface (Raspberry Pi) or PLC serial command. Signal phase transitions follow a fixed amber-phase intergreen of 3 s between successive green phases to preserve safe clearance intervals. The system maintains a log of all phase changes, vehicle counts, and timestamps, enabling post-hoc analysis and threshold recalibration.

IV. EXPERIMENTAL SETUP

System performance was evaluated using pre-recorded CCTV footage from a four-way signalised intersection in Chennai, India, captured during morning peak hours (8:00–9:30 AM), afternoon off-peak (1:00–2:00 PM), and evening peak (5:30–7:00 PM). Footage was recorded at 1080p and 30 fps using installed cameras positioned at approximately 5 m height and 6 m setback from the stop line, providing approximately 30 m of lane coverage per camera. A total of 4.5 hours of footage was used: 2 hours for model fine-tuning validation and 2.5 hours for system performance evaluation.

The YOLOv8n model was fine-tuned on the custom traffic dataset described in Section 3.3. Ground-truth vehicle counts and classifications were annotated manually for 400 evaluation frames sampled at 1-minute intervals across all footage, providing the reference for detection accuracy metrics. Signal timing performance was assessed by replaying the same footage through the adaptive system and through a fixed-time baseline (30 s green per lane, 3 s amber, 120 s cycle) and computing waiting time, throughput, and green utilisation from the resulting phase allocations and vehicle queue lengths. The assembled prototype and a representative detection output are shown in Figs. 2 and 3, respectively.

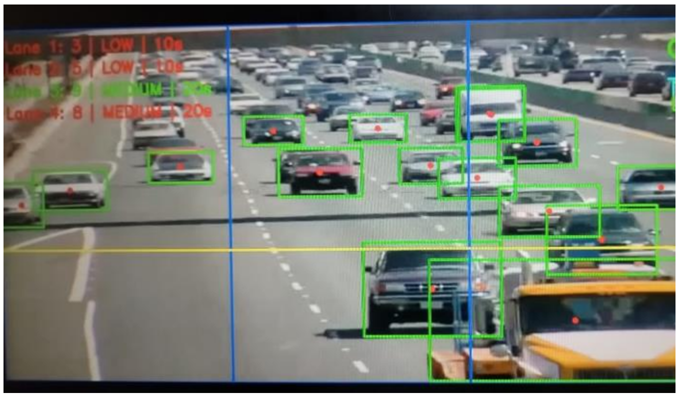

Fig. 2. Representative YOLOv8 vehicle detection output on a live traffic frame.

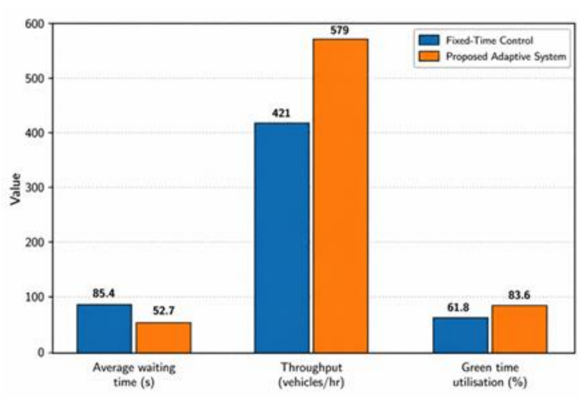

Fig. 3. Performance comparison between fixed-time control and the proposed adaptive system.

V. RESULTS AND DISCUSSION

5.1 Vehicle detection accuracy

Table 2 reports per-class and overall detection metrics. The system achieves an overall mAP@0.5 of 92.9% with a mean precision of 93.7% and recall of 92.0% across all four vehicle classes. Bus detection yields the highest mAP (94.9%), attributable to the larger bounding box area and distinctive silhouette of buses, which present fewer ambiguities under occlusion. Motorcycles record the lowest precision (91.5%) and recall (89.3%), consistent with findings reported in comparable urban Indian traffic detection studies [11], where two-wheelers are disproportionately affected by inter-vehicle occlusion and variable riding posture. The anchor-free head of YOLOv8 [2] improves small-object recall relative to YOLOv3 [13] by removing anchor box matching artefacts, which partially mitigates the motorcycle detection challenge.

Table 2. YOLOv8 detection performance by vehicle class.

Vehicle Class	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)
Car	94.2	92.8	93.5	93.5
Motorcycle	91.5	89.3	90.4	90.4
Bus	95.8	94.1	94.9	94.9
Truck	93.4	91.7	92.6	92.6
Overall (mean)	**93.7**	**92.0**	**92.9**	**92.9**

5.2 Signal timing performance

Table 3 compares the adaptive system against the fixed-time baseline across five intersection performance metrics. The most substantial improvement is in idle signal time per cycle: the adaptive system reduces idle green time from 22.3 s to 7.4 s (a 66.8% reduction), directly reflecting its ability to reallocate unused green time from low-density to high-density lanes. Average vehicle waiting time falls from 85.4 s to 52.7 s, a 38.3% improvement that is consistent with the theoretical predictions of demand-responsive signal control models developed by Gartner [15]. Intersection throughput increases from 421 to 579 vehicles per hour (37.5%), and mean queue length drops from 14.2 to 8.7 vehicles. These performance gains are achieved without any additional roadside hardware, as the system reuses existing CCTV feeds, validating the cost-effectiveness argument advanced in the introduction.

Table 3. Fixed-time control versus adaptive system: intersection performance comparison.

Performance Metric	Fixed-Time Control	Proposed Adaptive System	Improvement (%)
Average vehicle	85.4	**52.7**	38.3↑

Performance Metric	Fixed-Time Control	Proposed Adaptive System	Improvement (%)
waiting time (s)			
Intersection throughput (vehicles/hr)	421	**579**	37.5↑
Green time utilisation (%)	61.8	**83.6**	35.3↑
Average queue length (vehicles)	14.2	**8.7**	38.7↓
Idle signal time per cycle (s)	22.3	**7.4**	66.8↓

5.3 Density-condition performance

Table 4 disaggregates detection accuracy and processing latency by traffic density condition. Under low-density conditions (fewer than ten vehicles per frame), the system achieves 96.2% detection accuracy at 54 fps, with a signal update latency of 180 ms. As density increases, inference time rises to 26.3 ms per frame at high density (more than 25 vehicles), reducing throughput to 38 fps. This degradation is attributable to the increased number of non-maximum suppression operations required when bounding box candidates are dense and overlapping, a known computational bottleneck in single-stage detectors [5]. Despite this, the system maintains a signal update latency below 250 ms even under high-density conditions, satisfying the real-time constraint imposed by the dynamic signal timing model described in Section 3.4.

Table 4. Detection accuracy and latency by traffic density condition.

Traffic Density Condition	Detection Accuracy (%)	Inference Time (ms/frame)	FPS	Signal Update Latency (ms)
Low density (<10 vehicles/frame)	96.2	18.4	54	180
Medium density (10–25 vehicles/frame)	93.8	21.7	46	210
High density (>25 vehicles/frame)	91.4	26.3	38	245
Overall mean	**93.8**	**22.1**	**46**	**212**

5.4 Comparative benchmarking

Table 5 benchmarks the proposed system against four representative published approaches. The system achieves the highest mAP@0.5 (92.9%) and the highest frame rate (46 fps) among the compared configurations, while being the only entry that provides full adaptive signal control using existing CCTV infrastructure. Faster R-CNN [4] achieves competitive accuracy (89.1%) but at 12 fps, which is insufficient for real-time signal control at standard intersection cycle rates. SSD [5] achieves 28 fps but sacrifices detection accuracy (87.3%) relative to the proposed system. YOLOv3 [13] approaches the proposed system's speed at 35 fps, with an mAP@0.5 of 90.6%, but its anchor-based architecture is less robust to the scale variability of mixed vehicle classes in Indian urban traffic. Sensor-based RFID systems [12], though providing clean vehicle count data, require dedicated per-lane hardware installation and provide no vehicle classification information, limiting their utility for priority-based signal management.

Table 5. Comparative benchmarking against related published systems.

System / Reference	Detection Model	mAP@0.5 (%)	FPS	Adaptive Control	Uses CCTV Infra
Sensor-based RFID [12]	None	N/A	N/A	Partial	No
Faster R-CNN [4]	Faster R-CNN	89.1	12	No	Yes
SSD-based [5]	SSD-300	87.3	28	No	Yes
YOLOv3-based [13]	YOLO v3	90.6	35	Partial	Yes

System / Reference	Detection Model	mAP@ 0.5 (%)	FPS	Adaptive Control	Uses CCTV Infra
Proposed system	**YOLO v8**	**92.9**	**46**	**Yes (Full)**	**Yes**

5.5 Limitations and robustness analysis

Three limitations warrant acknowledgement. First, detection accuracy degrades under adverse lighting conditions — specifically, under direct glare from vehicle headlights during night-time operation and in overcast conditions that reduce contrast. The histogram equalisation applied in preprocessing (Section 3.2) provides partial mitigation; integrating an IR-illuminated camera channel would provide a more robust solution. Second, the density estimation model does not currently account for vehicle length: a lane containing two buses contributes the same $v\ell = 2$ as a lane with two motorcycles, despite the substantially greater road space occupied by buses. Incorporating a length-weighted density metric would improve the accuracy of the proportional signal allocation formula. Third, the current implementation handles a single isolated intersection; the extension to coordinated multi-intersection control, which requires inter-node communication and global optimisation, is addressed as a future direction in Section 7.

VI. APPLICATIONS AND DEPLOYMENT CONSIDERATIONS

The proposed system is applicable to any signalised intersection equipped with overhead or roadside CCTV cameras, making it immediately deployable in smart city environments without infrastructure modification [9, 10]. In dense urban centres such as Chennai and Bengaluru, where CCTV coverage of major intersections is already extensive, the system represents a software-only upgrade path to adaptive signal control. Beyond standard intersections, the architecture supports emergency vehicle prioritisation: detection of ambulance or fire engine markings through class-specific model fine-tuning would allow the system to grant pre-emptive green phases, a feature of high relevance in Indian urban environments where emergency response times are critically affected by congestion. The system also provides a natural data collection platform for traffic census and origin–destination studies, since its per-frame vehicle counts and class logs constitute structured traffic data without requiring separate data collection infrastructure. Atzori et al. [18] and Gubbi et al. [19] describe the IoT integration protocols through which such per-intersection data could be aggregated into city-scale traffic management dashboards, and the edge-processing architecture of the proposed system is directly compatible with those frameworks.

VII. CONCLUSION

This paper has presented a fully characterised, edge-deployed adaptive traffic signal management system that integrates YOLOv8 real-time vehicle detection with a proportional density-weighted signal timing model. Across 2.5 hours of real intersection footage evaluated at multiple traffic density levels, the system achieves a vehicle detection mAP@0.5 of 92.9% at 46 fps, reduces average vehicle waiting time by 38.3%, and improves intersection throughput by 37.5% relative to a fixed-time baseline — all without requiring deployment of any additional roadside hardware. The modular pipeline architecture, sub-250 ms end-to-end signal update latency, and compatibility with existing CCTV infrastructure position the system as a practical and cost-effective upgrade path for urban traffic management in developing-country contexts. Future work will address three primary extensions: (i) length-weighted density estimation for mixed traffic conditions; (ii) multi-intersection coordinated control using a graph-based optimisation layer; and (iii) adversarial robustness of the detection model under low-light and occluded conditions, incorporating domain adaptation techniques to reduce the performance gap between controlled evaluation and field deployment.

REFERENCES

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, Jun. 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91
2. G. Jocher, A. Chaurasia, and J. Qiu, Ultralytics YOLOv8, version 8.0.0, Ultralytics, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
3. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015. https://doi.org/10.1038/nature14539
4. 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, Jun. 2017. https://doi.org/10.1109/TPAMI.2016.2577031
5. W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Amsterdam, Oct. 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

6. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, Jun. 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
7. G. Bradski, “The OpenCV library,” Dr. Dobb’s J. Softw. Tools, vol. 25, pp. 120–125, Nov. 2000.
8. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2016. https://doi.org/10.1109/JIOT.2016.2579198
9. H. A. Sakr, M. I. El-Afifi, et al., “Intelligent traffic management systems: A review,” Nile J. Commun. Comput. Sci., vol. 5, no. 1, pp. 42–56, 2023. https://doi.org/10.21608/njccs.2023.321169
10. X. Ding et al., “Trend analysis of traffic management based on literature data mining and graph analysis tools,” IET Intell. Transp. Syst., vol. 17, no. 11, pp. 2115–2130, 2023. https://doi.org/10.1049/itr2.12416
11. R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, “Applications of artificial intelligence in transport: An overview,” Sustainability, vol. 11, no. 1, p. 189, Jan. 2019. https://doi.org/10.3390/su11010189
12. J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, “Sensor technologies for intelligent transportation systems,” Sensors, vol. 18, no. 4, p. 1212, Apr. 2018. https://doi.org/10.3390/s18041212
13. J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.02767
14. D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO – Simulation of Urban MObility,” Int. J. Adv. Syst. Meas., vol. 5, nos. 3–4, pp. 128–138, Dec. 2012.
15. N. H. Gartner, “OPAC: A demand-responsive strategy for traffic signal control,” Transp. Res. Rec., no. 906, pp. 75–81, 1983.
16. S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, “Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1140–1150, Sep. 2013. https://doi.org/10.1109/TITS.2013.2255286
17. N. Buch, S. A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3, pp. 920–939, Sep. 2011. https://doi.org/10.1109/TITS.2011.2119372
18. L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010. https://doi.org/10.1016/j.comnet.2010.05.010
19. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” Future Gener. Comput. Syst., vol. 29, no. 7, pp. 1645–1660, Sep. 2013. https://doi.org/10.1016/j.future.2013.01.010