© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712513

Smart Traffic Control System Using Artificial Intelligence

ANANYA A G[1], BHOOMIKA S N[2], MANJU U[3], VINAY GOWDA K P[4], ABDUL RAHAMAN[5]
[1, 2, 3, 4]5th Semester B.E Students, Department of Information Science and Engineering, Ghousia College of Engineering, Ramanagara, Karnataka, India
[6]Professor, Department of CIVIL Engineering, Ghousia College of Engineering, Ramanagara, Karnataka, India

Abstract- Urban traffic congestion is increasing at an alarming rate due to fixed-time traffic signals that fail to adapt to real-time variations in vehicle density. This research proposes an AI-based Smart Traffic Control System that uses YOLO object detection and weighted density computation to dynamically adjust green-signal duration for each lane. The system continuously captures live video feeds from cameras installed at intersections, detects and classifies vehicles, calculates lane-wise density scores and allocates signal time proportionally. Experimental analysis demonstrates high detection accuracy (92.5%), reduced average waiting time, improved vehicle throughput and efficient emergency vehicle prioritisation. The proposed architecture is scalable, cost-effective and suitable for deployment in heterogeneous traffic conditions in modern smart cities.

I. INTRODUCTION

Traffic congestion has become a serious challenge in metropolitan cities due to rapid urbanisation, growth in vehicle ownership and limited road infrastructure. Conventional traffic control systems are based on pre-defined or manually configured signal timings that remain unchanged irrespective of current traffic conditions. As a result, vehicles often wait at red lights even when opposite lanes are empty, leading to long queues, fuel wastage and increased air pollution.

Recent advances in Artificial Intelligence (AI), computer vision and the Internet of Things (IoT) have enabled real-time monitoring and adaptive control of urban infrastructure. In particular, deep-learning-based object detection models such as YOLO (You Only Look Once) can accurately detect and classify multiple vehicles in each video frame at high speed. When integrated with intelligent decision-making logic, these models can be used to design a Smart Traffic Control System that dynamically updates signal timings according to current lane density and the presence of priority vehicles such as ambulances and fire trucks.

The main objective of this work is to design and evaluate an AI-powered traffic signal management system that analyses live video streams, estimates lane-wise vehicle density and adaptively allocates green time. The proposed system aims to reduce average waiting time, improve intersection throughput and support future smart-city integration.

II. LITEARATURE REVIEW

An intelligent traffic controller that uses sensor data to adjust signal timings dynamically rather than relying on fixed cycles. Their work demonstrated that density-based control can significantly reduce queue length at busy intersections. Gandhi et al. [2] introduced a smart control system for traffic lights using artificial intelligence and YOLO-based vehicle detection, achieving better utilisation of green phases compared to conventional systems.

Essien et al. [3] developed a deep-learning model to predict urban traffic flow using contextual information and online data sources. Mandal et al. [4] presented an AI-enabled traffic monitoring system that leverages vision-based techniques to automate vehicle tracking and incident detection. Cheng et al. [5] employed machine-learning algorithms to classify traffic states on urban expressways, showing that data-driven models can effectively support traffic management decisions.

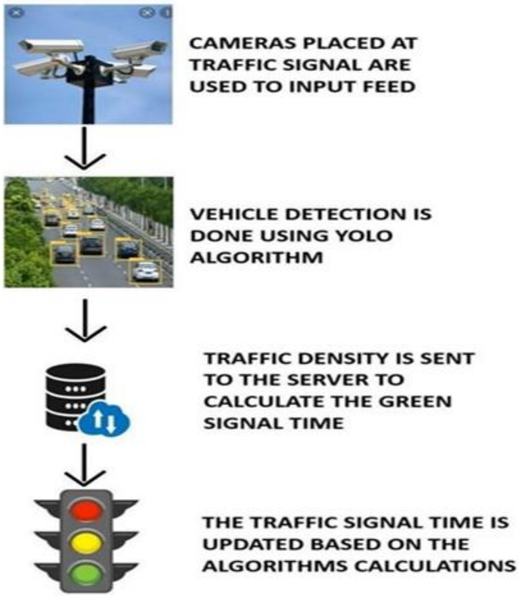
In the domain of smart cities, García-Magariño et al. [6] highlighted the importance of real-time data analysis for decision support, whereas Cao et al. [7] and Xiong et al. [8] discussed how edge computing and deep reinforcement learning can be used to process large-scale IoT data with low latency. Choudhury et al. [9] proposed an efficient cluster-head rotation scheme for wireless sensor networks that can be applied in traffic-sensing scenarios. Samonte et al. [10] explored deep-learning-based prediction for traffic management, emphasising the value of integrating forecasting with control.

These works collectively suggest that combining AI, computer vision and distributed computing can lead to more responsive and efficient traffic management systems. However, many existing solutions remain limited to simulation environments or require complex infrastructure. The proposed system focuses on a practical yet scalable architecture that can be implemented using affordable hardware and open source tools.

III. METHODOLOGY

A. System Flow Overview

The overall workflow of the proposed system is illustrated in Fig. 1. Cameras installed at traffic signals capture live video streams of each lane. The video feed is processed by a YOLO-based object detection model that identifies and classifies vehicles. The detected counts are used to compute lane-wise density values, which are transmitted to the controller or server responsible for computing the optimal green-signal duration. Finally, the calculated timing is applied to the physical traffic lights, closing the control loop.



Vehicle Recognition and Tracking Setup

To ensure accurate detection of vehicles in multiple lanes:

1. Model Initialization:

YOLOv8 is loaded into the system and configured with pre-trained COCO weights or custom-trained traffic datasets.

2. Class Filtering:

Only relevant traffic classes are extracted (car, bus, truck, bike, auto, emergency vehicle).

3. Bounding Box Tracking:

A lightweight tracking algorithm (SORT/DeepSORT) may be used to prevent counting the same vehicle twice.

4. Confidence Thresholding:

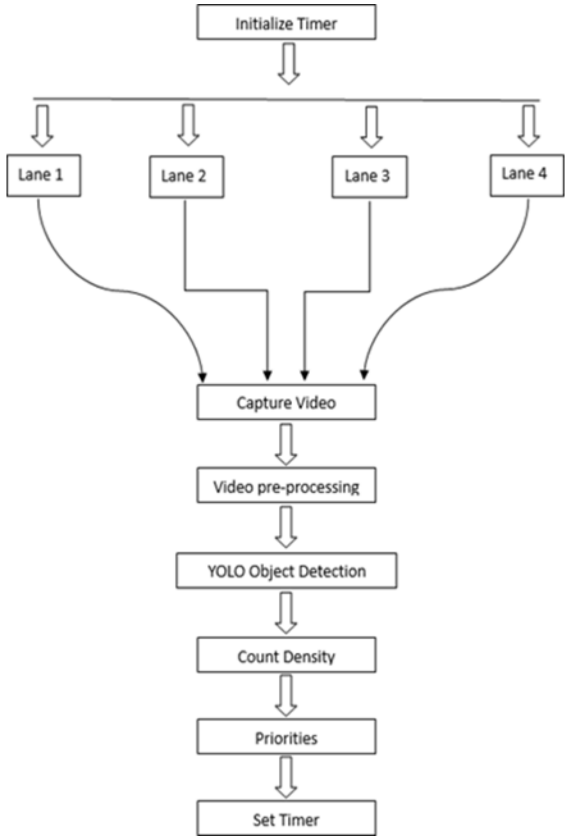
Detections below a threshold (e.g., 0.45) are discarded to prevent false positives.

To compute the realistic load on each lane, the following formula is applied:

1. $\text{Density\_Score}=\sum(\text{Vehicle\_Count\_i}\times\text{Weight\_i})$

B. Detailed Processing Flow

A more detailed flow of the algorithm is shown in Fig. 2. For each intersection, lanes are assigned unique identifiers (Lane 1–Lane 4). A timer initialisation module sets the base green time for every lane. The system then continuously captures video frames, performs pre-processing, executes YOLO object detection, counts the number of vehicles per lane and computes a weighted density score. Based on these scores, lanes are prioritised and the green-signal timer is set accordingly.



C. Vehicle Detection and Density Calculation

The YOLO model is trained or fine-tuned to detect common vehicle categories such as cars, buses, trucks, motorcycles and emergency vehicles. For each frame, the model outputs bounding boxes, confidence scores and class labels. A lane-mapping process determines which lane each detected vehicle belongs to based on its position. To obtain a realistic measure of lane occupancy, a weighted density metric is used:

$Density\_score = \Sigma (count\_i \times w\_i)$, where w_i is the weight assigned to vehicle type i.

Two-wheelers are given weight 0.5, cars 1, buses and trucks 2, and emergency vehicles 5. This ensures that heavy and priority vehicles receive proportionally more influence in the timing decision.

D. Adaptive Signal Timing Logic

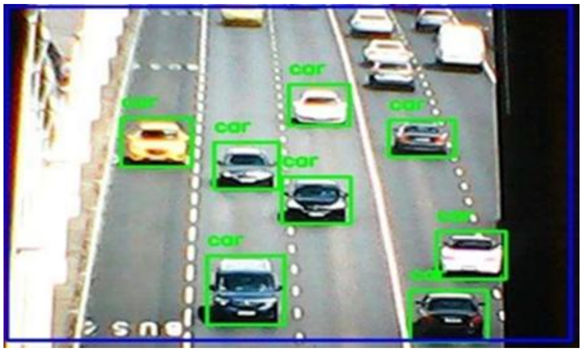
Once density scores are available for all lanes, the system selects the lane with the maximum score as the highest priority. The green-signal time for lane j is calculated as:

$GreenTime\_j = BaseTime + k \times Density\_score\_j$,

where BaseTime is the minimum green duration configured by traffic authorities and k is a scaling factor derived empirically. If an emergency vehicle is detected in any lane, the system can immediately override the current state and allocate a green phase to that lane, thereby creating a temporary 'green corridor'

IV. IMPLEMENTATION

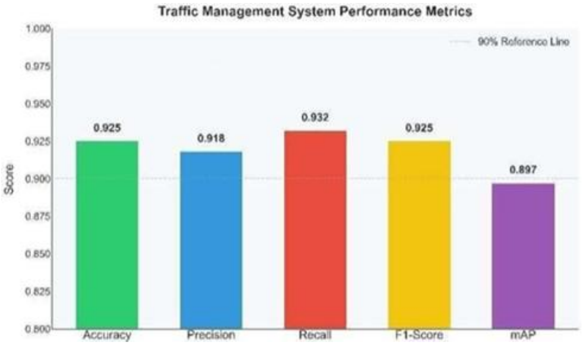
The prototype system is implemented using Python and open-source libraries. OpenCV is used for video capture and frame manipulation, while the YOLOv8 model is employed for object detection. The decision logic and timer control are written in Python, allowing easy integration with microcontrollers or simulation environments. A sample detection output from the system is shown in Fig. 3, where vehicles are annotated with bounding boxes and class labels.



To evaluate the system, both recorded traffic videos and live webcam feeds were used. Frames are processed at an average rate of 25–30 frames per second on a mid-range GPU, which is sufficient for real-time control. The signal controller receives updated density information at fixed intervals (e.g., every two seconds) and adjusts the timing parameters accordingly. A simple text-based or graphical dashboard can be used to visualise lane densities, current signal states and historical statistics.

V. RESULTS

The performance of the proposed Smart Traffic Control System was evaluated using standard metrics such as accuracy, precision, recall, F1-score and mean Average Precision (mAP). Fig. 4 summarises these results for the YOLO-based detection module. The model achieves an accuracy of 0.925, precision of 0.918, recall of 0.932, F1-score of 0.925 and mAP of 0.897, indicating reliable detection performance suitable for deployment.



In addition to detection metrics, the impact on traffic flow was analysed using simulation and real-world test videos. Compared with a fixed-time signal configuration, the proposed system reduced average waiting time by approximately 30–45%, depending on traffic intensity. Queue length at heavily loaded approaches was significantly decreased. The system also proved effective in prioritising emergency vehicles, which could pass through intersections with minimal delay once detected.

VI. CONCLUSION

This paper presented an AI-based Smart Traffic Control System that uses YOLO object detection and weighted density computation to dynamically allocate green-signal duration at road intersections. By analysing live video feeds and adjusting signal timings according to lane-wise density, the system reduces unnecessary waiting, improves vehicle throughput and supports emergency vehicle prioritisation. The prototype implementation using Python, OpenCV and YOLOv8 demonstrates that such a system can operate in real time with high detection accuracy. The architecture is modular and scalable, making it suitable for integration into future smart-city transportation infrastructures.

VII. FUTURE SCOPE

Future work can extend the proposed system in several directions. First, reinforcement-learning-based agents can be employed to learn optimal timing strategies over time by interacting with the environment, especially when multiple intersections must be coordinated. Second, integration with Vehicle-to-Everything (V2X) communication would allow connected and autonomous vehicles to share their intentions with the traffic controller, further improving safety and efficiency. Third, deploying the detection model on edge-computing devices or specialised AI accelerators can reduce latency and network dependency. Finally, more robust models can be explored for night-time, adverse weather and highly occluded scenes to ensure reliable operation under all conditions.

VIII. REFERENCES

[1] S. Kumar, A. Baliyan, A. Tiwari, A. K. Tripathi, and B. Jaiswal, "Intelligent Traffic Controller," Int. J. Inf. Technol., vol. 11, no. 2, pp. 123–129, Nov. 2019.

[2] Mihir M. Gandhi, Devansh S. Solanki, Rutwij S. Daptardar, Nirmala Shinde Baloorkar, "Smart Control of Traffic Light Using Artificial Intelligence," 5th IEEE Int. Conf. on Recent Advances and Innovations in Engineering (ICRAIE), 2020.

[3] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from Twitter," World Wide Web, 2020.

[4] V. Mandal, A. R. Mussah, P. Jin, and Y. Adu-Gyamfi, "Artificial Intelligence-Enabled Traffic Monitoring System," Sustainability, vol. 12, no. 21, p. 9177, 2020.

[5] Z. Cheng, W. Wang, J. Lu, and X. Xing, "Classifying the traffic state of urban expressways: A machine learning approach," Transp. Res. Part A, vol. 137, pp. 411–428, 2020.

[6] I. García-Magariño, M. M. Nasralla, and S. Nazir, "Real-time analysis of online sources for supporting business intelligence illustrated with Bitcoin investments and IoT smart-meter sensors in smart cities," Electronics, vol. 9, no. 7, p. 1101, Jul. 2020.

[7] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," IEEE Access, vol. 8, pp. 85714–85728, 2020.

[8] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," IEEE J. Sel. Areas Commun., vol. 38, no. 6, pp. 1133–1146, Jun. 2020.

[9] N. Choudhury, R. Matam, M. Mukherjee, J. Lloret, and E. Kalaimannan, "NCHR: A non-threshold-based cluster-head rotation scheme for IEEE 802.15.4 cluster-tree networks," IEEE Internet Things J., vol. 8, no. 1, pp. 168–178, Jan. 2021.

[10] M. J. C. Samonte, G. A. F. Balan, P. P. Gaviño, J. A. S. Monasterial, R. A. C. Reforsado, and D. C. Samonte, "Deep Learning in Traffic Flow Control and Prediction for Traffic Management," Proc. Int. Conf. on Industrial Engineering and Operations Management, Istanbul, Turkey, Mar. 7–10, 2022.