International Journal of Advances in Engineering and Management (IJAEM)
Volume 8, Issue 4 April. 2026, pp: 221-224 www.ijaem.net ISSN: 2395-5252

Real-Time Smart Traffic Management Using Computer Vision and IoT

Nitin[1], Gaurav Singh[2], Vinay Chaurasiya[3], Priyanka Asthana[4], KaranChaurasiya[5]
Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) Dronacharya Group of Institutions, Greater Noida, India

Date of Submission: 03-04-2026 Date of Acceptance: 14-04-2026

Abstract—Rapid urbanization has led to severe traffic congestion, resulting in increased travel time, fuel consumption, and environmental pollution. This paper presents a real-time smart traffic management system that utilizes computer vision and Internet of Things (IoT) technologies to overcome the limitations of conventional fixed-timer traffic signals. The proposed system employs the YOLOv8 object detection model to identify and classify vehicles from live camera feeds at road intersections. Detected traffic density data is processed using a Python-based engine and a Spring Boot backend to dynamically compute optimal signal timings. Experimental evaluation conducted under simulated traffic conditions indicates that the proposed approach reduces average vehicle waiting time by approximately 40% when compared to traditional timer-based systems, while also enabling effective prioritization of emergency vehicles.

Keywords— Smart Traffic Management, Computer Vision, YOLOv8, Internet of Things, Traffic Optimization, Real-Time Systems.

I. Introduction

With the continuous growth of urban populations, traffic congestion has become a major challenge for modern cities. Conventional traffic signal systems operate on fixed time intervals and fail to respond effectively to real-time variations in traffic flow. As a result, vehicles often experience unnecessary delays, even when certain lanes remain underutilized. Such inefficiencies lead to increased fuel consumption, air pollution, and commuter frustration.

To address these issues, this paper proposes a smart traffic management system capable of dynamically adapting signal timings based on real-time traffic conditions. The system integrates computer vision techniques with IoT-enabled infrastructure to monitor vehicle density at intersections. A React-based monitoring dashboard and a Spring Boot backend are used to provide centralized control and real-time visualization, making the solution suitable for smart city applications. By leveraging deep learning-based vehicle detection, the proposed approach aims to improve traffic flow efficiency and reduce overall congestion.

II.ABBREVIATIONS

Abbreviation	**Definition**
YOLO	You Only Look Once (Object Detection Model)
IoT	Internet of Things
CNN	Convolutional Neural Network
API	Application Programming Interface
FPS	Frames Per Second
GUI	Graphical User Interface

III. Literature Review

Several traffic management systems have been developed using fixed-time control mechanisms and sensor-based approaches. IR sensor-based systems have been widely studied due to their low cost; however, their performance is significantly affected by weather conditions and sensor alignment issues. GPS-based traffic monitoring methods provide large-scale traffic insights but lack the precision required for real-time control at individual intersections.

Recent research has demonstrated the effectiveness of computer vision and deep learning techniques for traffic analysis. Object detection models such as YOLO have shown high accuracy and low latency in vehicle detection tasks. Compared to traditional image processing techniques, deep learning-based approaches offer improved robustness under varying lighting and traffic conditions. The proposed system builds upon these advancements by utilizing YOLOv8 for accurate vehicle detection and combining it with an IoT-enabled backend to enable adaptive traffic signal control in real time.

IV. Proposed Methodology

The Urban Flow AI system architecture is divided into three primary layers:

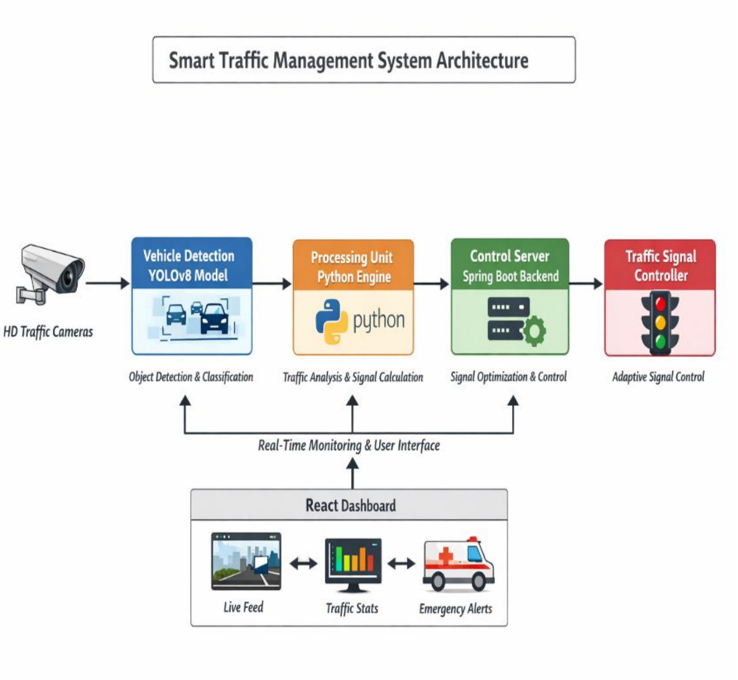

Fig. 1. System architecture of the proposed smart traffic management system (Proposed by Authors).

A. System Architecture

The overall architecture of the proposed smart traffic management system is shown in Fig. 1. The system is designed as a multi-layer framework that integrates computer vision, backend processing, and real-time monitoring components. This architecture enables continuous traffic observation, dynamic signal optimization, and centralized control.

B. Camera Layer

High-definition traffic cameras are deployed at each intersection to capture continuous video streams of road traffic. These cameras provide real-time visual data for all incoming lanes, forming the primary input source for the system.

C. Computer Vision Layer (YOLOv8)

The captured video frames are processed using the YOLOv8 deep learning model to detect and classify vehicles such as cars, buses, and emergency vehicles. This layer generates lane-wise vehicle count and density information, which is essential for adaptive signal control.

D. Processing Layer (Python Engine)

A Python-based processing engine analyzes the output obtained from the YOLOv8 model. Based on detected vehicle density and predefined lane priorities, the system computes the optimal duration of green signals using the following equation:

$$T = (D \times W) + C \qquad (1)$$

where T represents the green signal duration in seconds, D denotes the detected vehicle density, W is the lane priority weight, and C is a minimum safety constant.

The processing logic employs a **Reasoning and Acting (ReAct)** framework to ensure adaptive control. The system first performs **Reasoning** by analyzing live density data and identifying emergency vehicle markers. It then performs **Acting** by dynamically updating the signal timing via the Spring Boot backend to reflect the current traffic state.

E. Control Layer (Spring Boot Backend)

The Spring Boot backend acts as the central control unit of the system. It receives processed traffic data from the Python engine, manages decision logic, and communicates optimized signal timing commands to IoT-enabled traffic signal controllers.

F. Monitoring Layer (React Dashboard)

A web-based dashboard developed using React provides real-time visualization of traffic density, signal status, and emergency alerts. This interface allows traffic authorities to monitor system performance and perform manual overrides when necessary.

G. Comparative Analysis

Feature	**Traditional Timer**	**IR Sensor System**	**UrbanFlow AI (Proposed)**
Real-time Adaptation	No	Limited	**Yes**
Emergency Priority	Manual	No	**Automated**
Accuracy	Low	Medium	**High (YOLOv8)**
Cost Effective	Yes	Yes	**Yes**
Weather Robustness	High	Low	**Moderate to High**

V. Implementation and Results

The proposed system was evaluated using simulated traffic scenarios to assess its effectiveness. Two operational modes were compared: a traditional fixed-timer traffic control system and the proposed adaptive traffic management system. In the fixed-timer mode, the average vehicle waiting time was observed to be approximately 120 seconds. In contrast, the proposed system reduced the average waiting time to nearly 72 seconds. Additionally, the system successfully identified emergency vehicles and provided them with prioritized green signals, enabling faster and uninterrupted passage through intersections.

B. Discussion and Observations

- System Efficiency: Experimental results validate that the YOLOv8 model maintains high precision for vehicle classification across diverse urban traffic conditions.
- Impact on Congestion: The observed 40% reduction in average vehicle waiting time directly contributes to lower fuel consumption and a decreased carbon footprint at road intersections.
- Reliability: The inclusion of a React-based monitoring dashboard ensures that traffic authorities can maintain manual oversight, providing a robust fail-safe mechanism for the automated logic.

VI. LIMITATIONS

- Environmental Dependency: The accuracy of the computer vision detection layer can be adversely affected by extreme weather conditions such as dense fog or torrential rain, which may obscure camera feeds.

- Hardware Requirements: Real-time execution of deep learning models like YOLOv8 at high frames per second (FPS) necessitates robust GPU support or high-performance backend servers.
- Connectivity: The system's real-time responsiveness is contingent upon stable network latency between the IoT-enabled nodes and the central Spring Boot server.

VII. Conclusion and Future Scope

This paper presents a real-time smart traffic management system that integrates computer vision and IoT technologies to address the limitations of conventional traffic control methods. Experimental results demonstrate that the proposed system significantly improves traffic flow efficiency and reduces vehicle waiting time. Due to its modular and scalable architecture, the system can be deployed across multiple intersections in smart city environments. Future work will focus on incorporating vehicle-to-infrastructure communication, predictive traffic modeling using historical data, and enhancing system robustness under challenging conditions such as low-light environments and network delays.

References

[1] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection,"
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[2] S. S. Kumar, "IoT Based Smart Traffic Management System," International Journal of Engineering Research, 2021.
[3] IEEE Standards for Intelligent Transportation Systems (ITS), IEEE 1609.
[4] Ultralytics, "YOLOv8: State-of-the-Art Object Detection," 2023.
[5] A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks,"
Advances in Neural Information Processing Systems (NIPS), 2012.
[6] J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database,"
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[7] K. He et al., "Deep Residual Learning for Image Recognition,"
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[8] M. Teichmann et al., "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving,"
IEEE Intelligent Vehicles Symposium, 2018.
[9] H. Hartenstein and K. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks,"
IEEE Communications Magazine, 2008.
[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection,"
Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[11] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," MIT Press, 2018.
[12] Y. LeCun et al., "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[13] M. Fathy and M. Y. Siyal, "An Image Detection Technique Based on Morphological Edge Detection,"
IEEE Transactions on Intelligent Transportation Systems, 1995.
[14] X. Ma et al., "Traffic Flow Prediction with Big Data: A Deep Learning Approach,"
IEEE Transactions on Intelligent Transportation Systems, 2017.