WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

AI-Based Adaptive Traffic Management System

Prof. Vivek V.K.[1], Spoorti Pattar[2], Vidyashree Logavi[3], Kushal Teradal[4], Shrishail Atar[5]

[1] *Assistant Professor, Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi*
[2] *Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi*
[3] *Department of Electronics and communication Engineering/Jain College of Engineering and Research, Belagavi*
[4] *Department of Electronics and communication and Engineering/Jain College of Engineering and Research, Belagavi*
[5,] *Department of Electronics and communication and Engineering/Jain College of Engineering and Research, Belagavi*

ABSTRACT—

Traffic congestion at road intersections causes delays and difficulties for emergency vehicles. This paper presents an AI-Based Adaptive Traffic Management System that analyses a pre-recorded traffic video using OpenCV and YOLO to detect and count vehicles in individual lanes. Based on vehicle counts, traffic density is classified as low, medium, or high, and the analysed information is sent to an ESP8266 NodeMCU. The microcontroller controls two traffic signal models with red, yellow, and green LEDs, while two I2C LCD displays show traffic conditions and signal timing. When an ambulance is detected in a lane, the system gives that lane priority by activating the green signal and turning on a buzzer. A web dashboard supports hardware monitoring and control. The proposed system aims to improve traffic flow, reduce waiting time, and provide priority passage for emergency vehicles.

Index Terms— Adaptive traffic management, ESP8266 NodeMCU, OpenCV, YOLO, vehicle detection, traffic density, ambulance priority, traffic signal control.

I. INTRODUCTION

Traffic congestion is a major problem at road intersections, causing delays, fuel wastage, and increased pollution. Conventional traffic signals often use fixed timing and may not respond effectively to changing traffic conditions. An adaptive traffic management system can help improve traffic flow by adjusting signal timing according to traffic density.

The proposed AI-Based Adaptive Traffic Management System uses a pre-recorded traffic video as input. Open CV processes the video frames, while YOLO detects and counts vehicles in individual lanes. Based on the vehicle count, traffic density is classified as low, medium, or high.

The ESP8266 NodeMCU receives the analysed traffic information and controls two traffic signal models with red, yellow, and green LEDs. Two I2C LCD displays show vehicle counts, traffic density, and signal timing. A web dashboard is also used to monitor and control the hardware.

The system includes an emergency vehicle detection feature to give priority to ambulances. When an ambulance is detected in a lane, the corresponding traffic signal turns green according to the programmed control logic, and a buzzer activates to indicate the emergency. The system aims to reduce waiting time, improve traffic flow, and support faster passage for emergency vehicles.

II. LITERATURE SURVEY

Researchers have explored several approaches to intelligent traffic management, including computer vision, deep learning, vehicle counting, traffic density estimation, adaptive signal timing, and emergency vehicle detection. These techniques aim to reduce traffic congestion, improve signal efficiency, and provide priority passage for emergency vehicles.

Redmon et al. [1] introduced the YOLO object detection algorithm, which detects multiple objects in images in a single processing stage. Their work provides a foundation for detecting and counting vehicles in traffic videos, supporting AI-based traffic monitoring systems.

Alhazmi et al. [2] compared different YOLO models for adaptive traffic signal control using traffic videos. Their study examined detection accuracy, processing speed, and memory usage, highlighting the importance of selecting a suitable model for efficient vehicle detection and traffic analysis.

Khan et al. [3] proposed an adaptive traffic signal control system using object detection. Their research focuses on identifying vehicles and adjusting traffic signals according to traffic conditions, demonstrating the potential of computer vision for reducing unnecessary waiting times.

Bandi et al. [4] investigated a YOLO-based adaptive traffic signal control system with real-time traffic violation monitoring. Their approach uses vehicle detection and lane-wise counting to estimate traffic density and adjust signal timings, supporting the development of intelligent traffic management systems.

Chavan et al. [5] studied a computer vision-based adaptive traffic signal system using OpenCV and YOLO-based vehicle detection. Their work focuses on estimating traffic density and adjusting signal timings, while also considering emergency vehicle prioritization to improve traffic flow.

TABLE I. SUMMARY OF RELATED APPROACHES

Approach	**Strength**	**Limitation**
YOLO object detection [1]	Detects multiple objects efficiently in images	Does not directly control traffic signals
YOLO model comparison [2]	Helps evaluate vehicle detection accuracy and processing speed	Performance depends on the model and computing resources
Object detection-based adaptive control [3]	Supports vehicle detection and traffic-responsive signal timing	Requires reliable detection and signal-control logic
YOLO-based traffic control [4]	Supports vehicle counting and traffic violation monitoring	Detection performance may vary with traffic and lighting conditions
OpenCV and YOLO-based traffic management [5]	Supports traffic density estimation and adaptive signal control	Emergency priority and hardware integration require further validation
This work	Uses OpenCV, YOLO, ESP8266, two traffic signal models, two I2C LCDs, and a buzzer for density-based control and ambulance priority	Uses pre-recorded video; performance under real-time traffic conditions requires further testing

PROBLEM STATEMENT

Traffic congestion is a common problem at road intersections, especially during busy hours. Traditional traffic signals use fixed timing and may not adjust according to the number of vehicles on the road. This can lead to long waiting times, fuel wastage, and increased pollution.

Another major issue is the delay faced by ambulances at traffic signals. To address these problems, our project uses OpenCV and YOLO to detect and count vehicles from a pre-recorded traffic video. The system identifies traffic density as low, medium, or high and uses an ESP8266 NodeMCU to control the traffic signals. When an ambulance is detected, the system gives priority to its lane by turning on the green signal and activating a buzzer.

OBJECTIVES

The main objectives are:

(a) To analyse pre-recorded traffic videos using OpenCV and YOLO for vehicle detection and counting.

(b) To classify traffic density as low, medium, or high based on the number of vehicles in each lane.

(c) To use the ESP8266 NodeMCU to control two traffic signal models according to the detected traffic conditions.

(d) To display vehicle counts, traffic density, and signal timing on two I2C LCD displays.

(e) To detect emergency vehicles, particularly ambulances, and give priority to the lane in which they are detected.

(f) To activate the corresponding green signal and buzzer when an ambulance is detected.

(g) To develop a low-cost and efficient traffic management system that helps reduce waiting time, improve traffic flow, and support faster movement of emergency vehicles.

V. PROPOSED SYSTEM

The proposed system uses OpenCV, YOLO, and ESP8266 NodeMCU to analyse traffic conditions and control traffic signals. It uses a pre-recorded traffic video to detect and count vehicles in each lane.

A. System Architecture

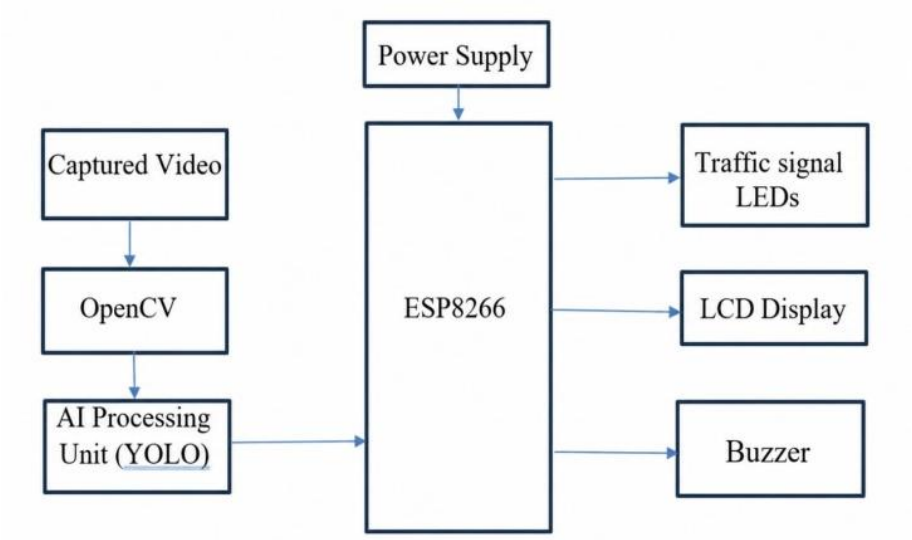
The proposed AI-Based Adaptive Traffic Management System uses OpenCV and YOLO to analyse a pre-recorded traffic video. OpenCV processes the video frames, while YOLO detects and counts vehicles in each lane. Based on the vehicle count, traffic density is classified as low, medium, or high. This information helps the system decide how to control the traffic signals according to the detected traffic conditions.

The analysed traffic information is sent from the laptop to the ESP8266 NodeMCU, which controls two traffic signal models with red, yellow, and green LEDs. Two I2C LCD displays show the vehicle count, traffic density, and signal timing. A web dashboard is also provided to monitor traffic conditions and control the hardware, making the system easier to operate.

The system also includes an ambulance detection feature to support emergency vehicle prioritization. When an ambulance is detected in a lane, the system gives priority to that lane by activating the corresponding green signal according to the programmed control logic. At the same time, the buzzer turns on to indicate the emergency. The laptop, ESP8266 NodeMCU, traffic signals, LCD displays, buzzer, and web dashboard work together to demonstrate the proposed traffic management system.

Overall, the system combines computer vision and embedded hardware to demonstrate traffic density-based signal control and ambulance prioritization. It aims to reduce unnecessary waiting time, improve traffic flow, and support faster movement of emergency vehicles. Further testing with real-time traffic videos and actual road conditions would help evaluate its performance in practical applications.

Fig.1. Block diagram of the proposed system



B. Key Components

ESP8266 NodeMCU: The main microcontroller used to control the traffic signal models. It receives the analysed traffic information and controls the signal lights based on traffic density and ambulance priority.

Laptop: Used to process the pre-recorded traffic video using OpenCV and YOLO. It detects and counts vehicles in each lane and identifies ambulances.

OpenCV and YOLO: OpenCV processes the video frames, while YOLO detects vehicles and ambulances. The vehicle count is used to classify traffic density as low, medium, or high.

Traffic Signal Models: Two traffic signal models, each with red, yellow, and green LEDs, represent traffic signals for two lanes. Their lights change according to the programmed control logic.

I2C LCD Displays: Two LCD displays show vehicle counts, traffic density, and signal timing, helping users monitor the traffic conditions.

Buzzer: Provides an audible warning when an ambulance is detected, indicating that emergency vehicle priority has been activated.

Web Dashboard: Provides an interface to monitor traffic information and control the hardware system.

Wi-Fi Communication: The ESP8266 uses Wi-Fi to communicate with the web dashboard and exchange information as configured in the system.

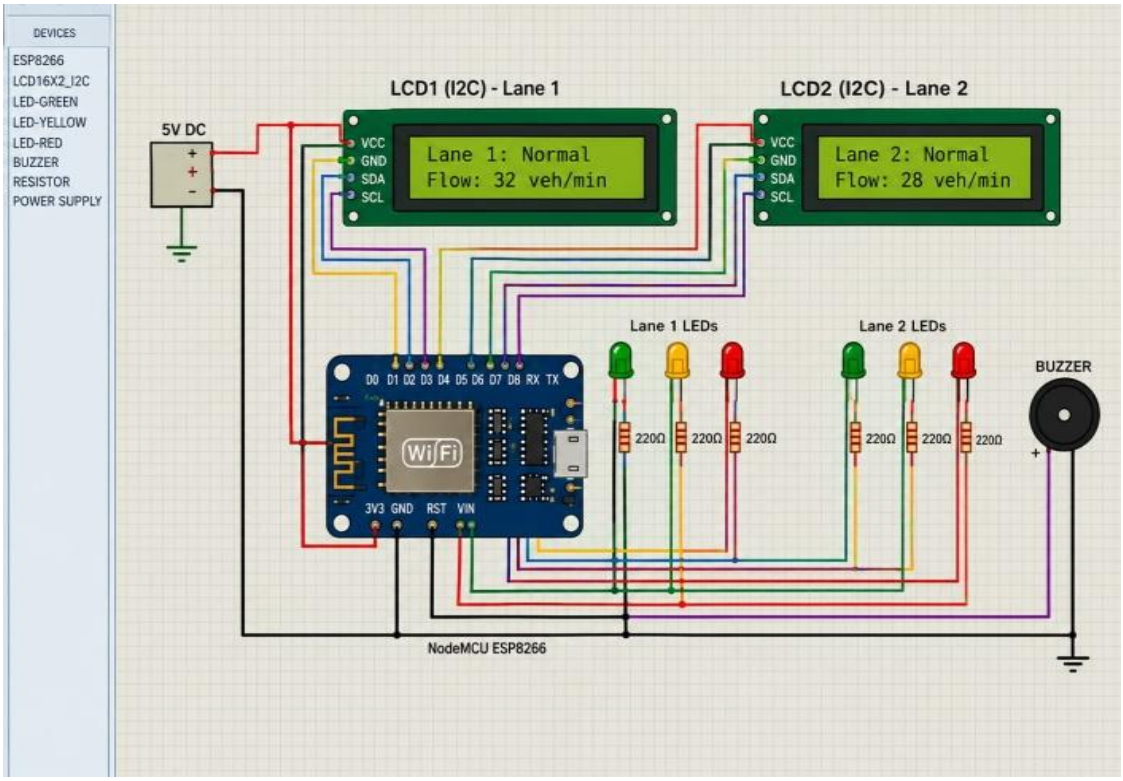
VI. IMPLEMENTATION

A. Hardware Implementation

The hardware prototype uses an ESP8266 NodeMCU to control two traffic signal models. Each model contains red, yellow, and green LEDs to represent traffic signal conditions. The system receives traffic information processed by a laptop and changes the signal lights according to the detected traffic density and programmed control logic.

Two I2C LCD displays are connected to show vehicle counts, traffic density, and signal timing. When an ambulance is detected in a lane, the system gives priority to that lane by activating the corresponding green signal and turning on the buzzer to indicate the emergency. A web dashboard is also used to monitor and control the hardware. Fig. 2 shows the hardware connection layout.

Fig.2. Circuit connection diagram



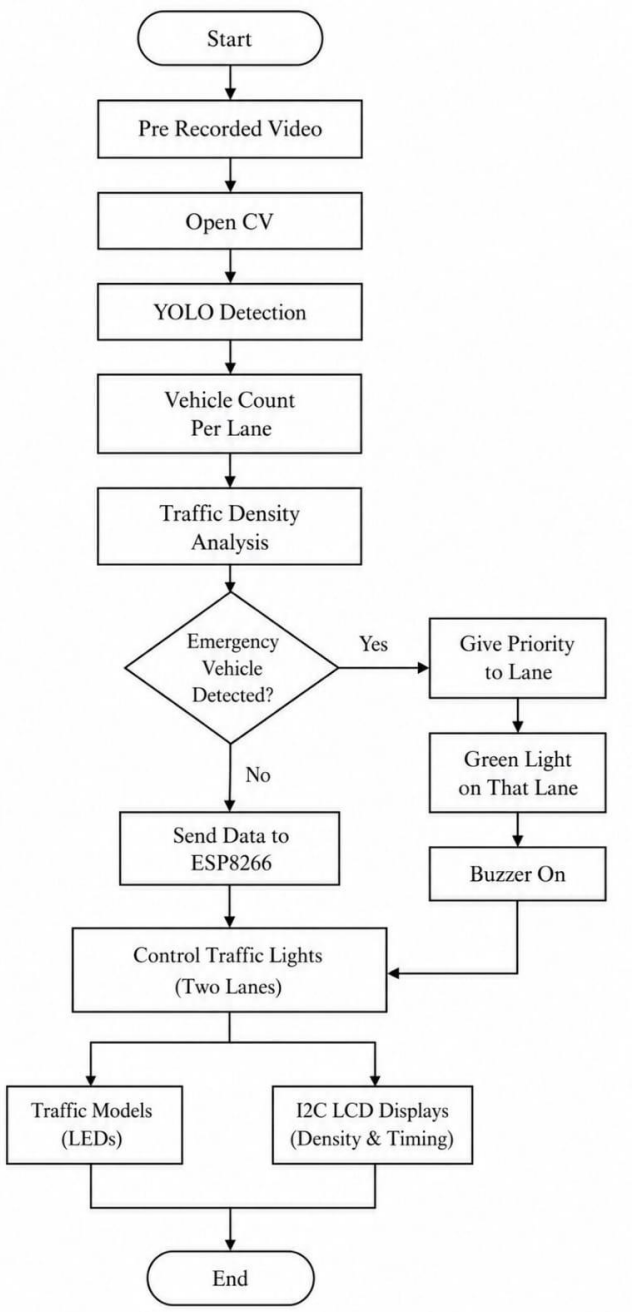
C. Software Implementation

The software implementation uses Python, OpenCV, and YOLO to analyse a pre-recorded traffic video. The ESP8266 NodeMCU is programmed to control the traffic signal models, LCD displays, and buzzer according to the analysed traffic information.

The system detects and counts vehicles in each lane, classifies traffic density as low, medium, or high, and adjusts the signal operation based on the programmed logic. The overall process follows Video Input → Vehicle Detection → Traffic Density Estimation → Signal Control, as shown in Fig. 3.

1. Load the pre-recorded traffic video and initialize the OpenCV and YOLO models.
2. Process the video frames and detect vehicles in each lane.
3. Count the detected vehicles and classify traffic density as low, medium, or high.
4. Send the analysed traffic information to the ESP8266 NodeMCU.
5. Update the traffic signal LEDs and LCD displays according to the detected traffic conditions.
6. If an ambulance is detected, give priority to its lane by activating the corresponding green signal and buzzer.
7. Continue processing the video and updating the traffic signals and dashboard as the system operates.

Fig.3. Flow diagram



C. Traffic Density Threshold Selection

Selecting suitable thresholds is an important part of the proposed traffic management system. The vehicle count may vary depending on the traffic video, lane conditions, and vehicle movement. Therefore, suitable vehicle-count ranges are defined to classify traffic density as low, medium, or high.

The process begins by analysing the pre-recorded traffic video and counting vehicles in each lane. The detected vehicle counts are then used to define density thresholds, which are tested and adjusted to obtain suitable results. Based on the classified traffic density, the system controls the signal timing through the ESP8266 NodeMCU. Ambulance detection is handled separately to give priority to the lane where the ambulance is identified. Repeated testing helps improve the reliability of vehicle detection and traffic signal control.

VII. RESULTS AND DISCUSSION

The proposed system uses OpenCV and YOLO to detect and count vehicles from a pre-recorded traffic video. Based on the vehicle count, traffic density is classified as low, medium, or high, and the ESP8266 NodeMCU controls the two traffic signal models accordingly.

The two LCD displays show vehicle counts, traffic density, and signal timing. When an ambulance is detected, the system gives priority to its lane by activating the green signal and turning on the buzzer. The web dashboard helps monitor and control the hardware.

The results demonstrate the basic working of traffic density-based signal control and ambulance prioritization. Further testing is required to evaluate vehicle detection accuracy, signal response, and overall system performance.

TABLE II. RECOMMENDED TESTS (RESULTS TO BE RECORDED)

Test Condition	**Expected Decision**	**Measured Result**
Low traffic density	Signal operates according to the programmed timing	To be measured
Medium traffic density	Signal timing adjusts according to vehicle count	To be measured
High traffic density	Higher-density lane receives appropriate signal priority	To be measured
Ambulance detected in a lane	Corresponding green signal activates and buzzer turns on	To be measured
LCD display operation	Vehicle count, traffic density, and timing are displayed	To be measured
Web dashboard operation	Traffic information and hardware controls function as intended	To be measured
Communication between laptop and ESP8266	Analysed traffic information is received correctly	To be recorded

The ambulance priority feature adds an important safety benefit by allowing the system to give a green signal to the lane where an ambulance is detected. The buzzer provides an audible indication of the emergency, while the web dashboard helps monitor the system. Since the project uses a pre-recorded traffic video, its performance under real-time road conditions has not yet been verified. Further testing should evaluate vehicle detection accuracy, ambulance detection accuracy, signal response time, communication reliability, and false detections.

VIII. ADVANTAGES AND LIMITATIONS

A. Advantages

The proposed system detects and counts vehicles from a pre-recorded traffic video and adjusts traffic signals according to traffic density. The ESP8266 NodeMCU controls the signal models, while the LCD displays provide information about vehicle counts, traffic conditions, and signal timing.

The ambulance priority feature helps provide faster passage for emergency vehicles by activating the corresponding green signal and buzzer. The web dashboard supports monitoring and control of the hardware. The system uses a compact, low-cost setup and demonstrates how AI-based vehicle detection can support more efficient traffic management.

B. Limitations

The system uses a pre-recorded traffic video, so its performance may differ from real-time road conditions. Vehicle and ambulance detection accuracy depends on video quality, lighting conditions, and the YOLO model. The system also depends on reliable communication between the laptop and ESP8266 NodeMCU. The traffic density thresholds and signal timing may require further adjustment through testing. Before real-world use, the system needs additional testing under different traffic conditions to ensure reliable operation.

IX. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper presents an AI-Based Adaptive Traffic Management System using OpenCV, YOLO, and ESP8266 NodeMCU. The system detects and counts vehicles from a pre-recorded traffic video, classifies traffic density, and controls two traffic signal models accordingly. The LCD displays provide traffic information and signal timing, while the ambulance priority feature activates the corresponding green signal and buzzer when an ambulance is detected.

The web dashboard supports hardware monitoring and control. The proposed system demonstrates the potential of AI-based traffic management to improve traffic flow and reduce waiting time. Further testing with real-time traffic videos is required to evaluate its accuracy and reliability under different road conditions.

B. Future Scope

The system can be improved by using real-time traffic video instead of pre-recorded video. More advanced AI models can be used to improve vehicle and ambulance detection accuracy under different lighting and traffic conditions. The system can also be extended to manage more traffic lanes and intersections with adaptive signal timing. A mobile application or cloud-based platform could be developed for remote monitoring and traffic data storage. In the future, integration with real traffic signals and emergency services could help improve traffic flow and provide faster passage for ambulances.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.

[2] A. S. Alhazmi, A. Hamza, and A. Moubayed, "Edge-Enabled Detection for Adaptive Traffic Signal Control: A Comparative Study of YOLO-Based Models in Intelligent Transportation Networks," Transportation Research Procedia, vol. 97, pp. 55–62, 2026.

[3] I. Khan, M. M. Hasan, S. Islam, M. S. Reza, M. T. Reza, and E. T. Efaz, "Adaptive Traffic Signal Control System Using Object Detection Based Approach," in Proceedings of the 2025 4th International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2025.

[4] W. Yu et al., "State-of-the-Art Review on Traffic Control Strategies for Emergency Vehicles," IEEE Access, vol. 10, pp. 109729–109742, 2022.

[5] A. Chavan, B. Dalvi, P. Gajjar, S. Jaiswal, and A. J. Striney, "Computer Vision-Based Adaptive Traffic Signal Control With Emergency Vehicle Prioritization," International Journal of Science, Engineering and Technology, 2026.

[6] "Adaptive Traffic Signal Control System Using Object Detection Based Approach," in Proceedings of the 2025 4th International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2025.

[7] L. Long et al., "A Collaborative Control Framework: Achieving Emergency Vehicle Priority While Minimizing Negative Impact on Ordinary Vehicles at Signalized Intersection," Journal of Advanced Transportation, vol. 2024.