© June 2025 | IJIRT | Volume 12 Issue 1 | ISSN: 2349-6002

Autonomous Traffic Signal Prioritization for Efficient Emergency Vehicle Routing

Nikhil Prakash Shinde[1], Harshita Chandrakant Soni [2], Aniket Adhik Mane[3], Mr.Vivekanand Shere[4]
[1,2,3] *Department of Electronics and Telecommunication Engineering, D Y Patil College of Engineering, Akurdi, India*
[4] *Assistant Professor, Department of Electronics and Telecommunication Engineering, D Y Patil College of Engineering, Akurdi, India*

***Abstract*—Rapid and unhindered movement of emergency vehicles is essential for saving lives, yet conventional traffic systems often obstruct this by failing to prioritize them at intersections. This paper introduces an intelligent, autonomous solution aimed at ensuring timely passage for ambulances and similar vehicles through urban traffic. By incorporating edge computing, real-time wireless communication, and localized signal control, the system offers a low-cost, practical alternative to existing infrastructure-heavy methods. A vision-enabled ESP32-CAM module serves as the core detection unit, identifying emergency vehicles within a predefined range based on visual indicators. Upon detection of an emergency vehicle, NRF24L01 transceiver forwards a wireless signal to a centralized ESP8266 controller, which orchestrates traffic signal behavior using the ESP-NOW protocol. This dynamically enables a green light for a fixed interval in the vehicle's direction while halting cross traffic.Simultaneously, a GPS-based alert mechanism transmits the vehicle's location to nearby drivers. A secondary ESP32 module processes this data and triggers alerts on a TFT screen within the car, providing early notice of an approaching emergency vehicle. The system eliminates the reliance on cloud connectivity or centralized servers, ensuring minimal latency and uninterrupted functionality. Designed for easy integration into smart city ecosystems, this approach demonstrates improvements in emergency response efficiency, road safety, and overall traffic fluidity. Its adaptability makes it a viable option for modernizing urban transportation networks.**

I. INTRODUCTION

In today's rapidly urbanizing world, traffic congestion has become one of the most persistent challenges, particularly in densely populated cities. While traffic delays are a daily inconvenience for regular commuters, they pose a far more serious problem for emergency vehicles such as ambulances, fire trucks, and police cars. In emergency situations, where every second matters, delays caused by red signals or traffic jams can be the difference between life and death. Traditionally, traffic signal systems are fixed and time-based, lacking the flexibility to adapt in real time to the presence of an emergency vehicle. Even in modern adaptive systems, emergency vehicle prioritization is often not integrated, leading to unpredictable delays and unsafe conditions. Moreover, most drivers only become aware of an approaching emergency vehicle when they hear a siren or see flashing lights—by which time it may already be too late to safely clear the path. This research presents a smart, cost-effective, and decentralized solution that provides real-time priority to emergency vehicles by dynamically controlling traffic signals and alerting nearby drivers. The system utilizes edge computing with ESP32-CAM for visual detection, wireless communication using ESP-NOW and NRF24L01 modules, and GPS-based proximity alerts displayed inside regular vehicles. Unlike cloud-dependent or infrastructure-heavy systems, the proposed approach works on local processing and wireless modules, ensuring low latency, high reliability, and easy scalability. It is specifically designed to be implemented using existing traffic setups with minimal modifications, making it a practical option for both developing and developed urban regions. By combining real-time detection, intelligent signaling, and driver notification, this system aims to reduce response times, prevent accidents, and enhance overall public safety in emergency scenarios.

II. LITERATURE REVIEW

Urban mobility and emergency response times are increasingly challenged by rising traffic congestion and inadequate signal prioritization at intersections. Traditional systems follow fixed signal cycles and do not recognize the dynamic need to accommodate emergency vehicles such as ambulances and fire trucks. To address this limitation, several researchers have explored the use of wireless communication, embedded systems, computer vision, and intelligent algorithms to improve the responsiveness of traffic infrastructure. This section reviews five significant works that have contributed to the development of smart traffic systems and their application in emergency vehicle routing.

[1]. Kouvelas et al. proposed an adaptive signal control system based on the Webster method for traffic signal optimization at isolated junctions. Their system dynamically calculates green signal times depending on the detected traffic density, making it more effective than traditional fixed-cycle traffic controllers. The system was tested in simulated saturated traffic conditions, and results showed reduced waiting times and improved flow efficiency. While their work addressed congestion using real-time data, the model was not designed to prioritize emergency vehicles. The system focused primarily on overall traffic flow and did not incorporate detection or preemption logic required for emergency scenarios. It served as a foundational framework for dynamic signal control but lacked integration with vehicle-specific inputs such as camera feeds or wireless modules.

[2]. In their work on cooperative traffic systems, Bauza et al. introduced the CoTEC (Cooperative Traffic Congestion Detection) method, aimed at enhancing vehicular communication and congestion management. The CoTEC system made use of vehicle-to-infrastructure (V2I) communication where connected vehicles shared their speed, location, and traffic data with roadside units. This collaborative data sharing helped detect congestion in real time and enabled rerouting recommendations. The performance of this method was evaluated using iTETRIS, an open-source simulation platform developed specifically for intelligent transportation systems. While CoTEC showed improvement in managing congestion through cooperation between vehicles and infrastructure, it did not include mechanisms for real-time detection or prioritization of emergency vehicles. The approach was comprehensive for general traffic control but lacked a dedicated strategy for emergency mobility. It also relied heavily on network coverage, limiting its application in areas with insufficient roadside communication infrastructure.

[3]. Zhao et al. focused on a vision-based traffic control system using low-cost image sensors and GPS for urban navigation. In their system, webcams installed at traffic junctions captured live images of road conditions. These images were then analyzed using a masking algorithm to estimate vehicle density, which in turn guided traffic signal duration. The system used sliding window smoothing estimators for accurate positioning and navigation in dense urban areas. Unlike traditional systems, this model leveraged visual information rather than physical sensors embedded in roads. Though the system demonstrated success in adjusting signals based on traffic density, its heavy reliance on cloud-based servers for image processing introduced considerable latency. In emergency use cases, where instant response is essential, this delay was seen as a significant disadvantage. Moreover, the system lacked the ability to directly identify or respond to emergency vehicles. There was no specific emergency detection algorithm, and the model required consistent internet connectivity, making it less reliable in areas with unstable networks.

[4]. A more centralized cloud-based approach was introduced by Mohapatra and Koundinya, who proposed an ambulance coordination system using GPS and real-time cloud computing. In their model, the ambulance location was tracked continuously and transmitted to a central server. The cloud platform then analyzed traffic conditions and provided optimized routes to ensure minimum delay. The system also allowed hospitals and traffic departments to monitor emergency vehicle movement, enhancing coordination during critical events. While the solution showed promise in improving the operational visibility of ambulance services, it was heavily dependent on cloud services for route processing and decision-making. In the absence of stable internet connectivity, the system's functionality was severely compromised. Furthermore, the model did not include a local traffic signal override mechanism or driver notification system. It relied

entirely on GPS-based rerouting, which could still face real-world challenges such as blocked lanes or signal delays. The lack of hardware integration with traffic controllers limited its practical deployment in cities with existing signal infrastructures.

[5]. Nida Farheen et al. proposed a density-based traffic light control system that included emergency vehicle detection using audio signals, particularly sirens. The system used IoT components such as microphones and sound sensors placed at intersections. When a siren was detected, the system automatically switched the corresponding traffic signal to green. This was one of the early models to include an emergency response mechanism in real time. However, the system had several limitations. Sound-based detection proved unreliable in high-noise environments such as markets or during heavy traffic. The sensors often misinterpreted horn sounds or ambient noises as sirens, leading to false triggers. Moreover, the range of detection was limited to the sound's reach, and vehicles without sirens—such as undercover police units—could not be identified. The system also did not include any alert mechanism for regular drivers, meaning vehicles in the path of the emergency response often reacted too late. While the concept demonstrated a novel idea, it lacked robustness and accuracy required for a critical application like emergency routing.

III.METHODOLOGY

The proposed system is designed to autonomously prioritize emergency vehicles by intelligently controlling traffic signals and alerting nearby drivers. It integrates real-time image processing, wireless communication, and GPS tracking to detect emergency vehicles, switch traffic signals accordingly, and display alerts inside regular vehicles. The system primarily consists of an ESP32-CAM module for edge-based visual detection, NRF24L01 transceivers for wireless data transmission, and ESP8266 microcontrollers configured in a Master–Slave setup for traffic signal control. Additionally, GPS modules are used to calculate the distance between the emergency vehicle and nearby cars, while a TFT display installed inside regular vehicles is used to visually notify drivers when an ambulance is approaching. All processing and communication are done locally, ensuring fast and reliable operation without reliance on cloud infrastructure. The system's modular design allows for decentralized control of each signal pole, making it adaptable for real-world deployment across multiple intersections in smart cities.

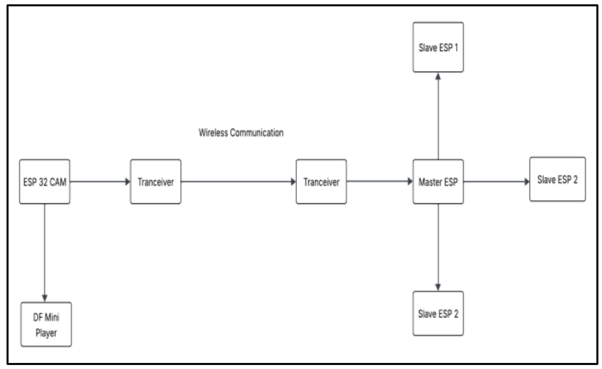

Fig 1. Block Diagram of Autonomous Traffic Control System

Functions of Blocks:

ESP32-CAM: The ESP32-CAM module acts as a smart detection unit. It is responsible for capturing visual input from the road and identifying the presence of emergency vehicles based on their visual markers, such as sirens, flashing lights, or body graphics. Once such a vehicle is identified, the module instantly sends a detection signal, initiating the control sequence.

DFPlayer Mini Module: Connected to the ESP32-CAM, this audio module plays a key role in generating spoken or sound-based warnings. For example, once an emergency vehicle is detected, the DFPlayer can play a preloaded sound—such as a beep or a verbal announcement—alerting pedestrians and nearby drivers of the approaching vehicle. It adds an extra layer of awareness at intersections.

Transceiver Modules: A pair of transceivers ensure seamless wireless communication between the ESP32-CAM and the central controller (Master ESP). These modules transmit and receive data packets, helping relay emergency detection information quickly and without delay. This enables real-time decision-making for signal adjustments.

Master ESP: The Master ESP module operates as the decision hub for traffic control. Upon receiving a signal from the ESP32-CAM via the transceiver, it determines the direction from which the emergency vehicle is arriving. Based on this, it issues commands to specific Slave ESPs to temporarily change their signal lights—green for the emergency lane and red for others. It handles the logic of the entire signal network.

Slave ESP Modules (1, 2, 3...): Installed at each traffic junction, these modules carry out instructions from the Master ESP. Each slave controls a traffic signal pole and is capable of switching red, yellow, and green LEDs based on received commands. These modules also control countdown displays if used, and they ensure that the signals operate as intended during normal and emergency modes.

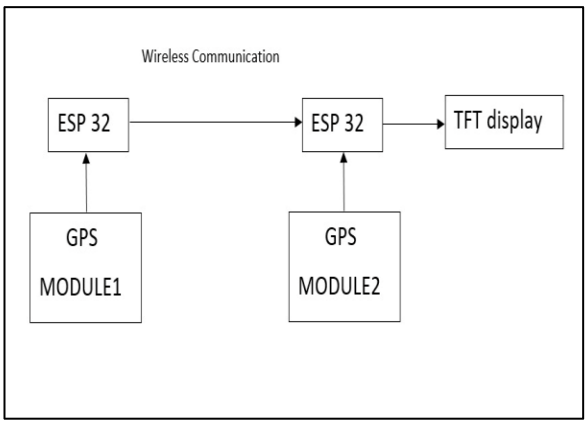

Fig.2. Block Diagram of Ambulance Detection and Driver Alert System

Functions of Blocks:

GPS Module 1 (Ambulance Unit): Positioned in the emergency vehicle, this GPS module continuously gathers location data and updates the microcontroller on its current coordinates. It provides the real-time position of the ambulance, which is then used to inform other parts of the system.

ESP32 (Ambulance Side): This ESP32 is tasked with reading GPS data and sending it out wirelessly. It broadcasts the ambulance's live coordinates over a dedicated channel, which are then picked up by nearby civilian vehicles to determine proximity.

GPS Module 2 (Vehicle Side): This module is used to fetch the civilian car's location in real time. The position is continuously updated and sent to the onboard ESP32, which uses this data to measure how close the ambulance is at any given moment.

TFT Display (Inside the Vehicle): Mounted on the dashboard, this display acts as the alert interface for the driver. If the calculated distance between the ambulance and the car is less than the predefined threshold (e.g., 100 meters), the screen shows a warning like "Ambulance Nearby – Give Way." This gives the driver enough time to react and assist in clearing the road.

1.Software Required:

- Arduino IDE: This development platform was used to program the ESP32-CAM and ESP8266 boards. Its built-in tools, such as the serial monitor and real-time compiler, made it easier to write and debug code. The IDE allowed us to integrate various libraries essential for handling the different components—such as LEDs, GPS modules, and communication protocols like ESP-NOW. It acted as the central workspace for programming logic related to traffic signal switching and data exchange between modules.
- Wokwi Simulator: To test the design virtually before moving to hardware, we used Wokwi—a browser-based simulation tool. It supports the simulation of ESP boards and allowed us to validate traffic signal timing, LED behavior, and communication between modules. Using this tool helped minimize hardware testing errors and saved development time by spotting bugs early in the design phase.
- Edge Impulse Studio: This tool was employed to train a visual recognition model that could identify emergency vehicles. The model was optimized for the ESP32-CAM so that it could perform real-time detection directly on the device. Using Edge Impulse allowed us to create a compact and efficient model that didn't rely on internet access or cloud servers, ensuring fast, local decision-making.
- Proteus Design Suite: Proteus was used to simulate the electronic components of the system, including microcontrollers, LEDs, and transceivers. It helped us test the behavior of the entire circuit in different conditions—especially how traffic signals would respond to incoming data. This ensured that the physical implementation was based on a tested and verified digital prototype.

2. Hardware specifications:

- ESP32-CAM: This microcontroller includes a camera and Wi-Fi capabilities, making it ideal for our vision-based detection system. It was positioned roadside to visually monitor oncoming traffic. The onboard camera worked alongside the deployed machine learning model to detect emergency vehicles without needing external servers. Its ability to process images locally helped us achieve a fast and efficient detection mechanism.
- ESP8266 Boards (Master and Slave): These boards were used to control the traffic lights. One ESP8266 unit was set up as the main controller (master), and it coordinated with several slave modules—each responsible for one signal pole. Communication between them was handled using ESP-NOW, which allowed direct data sharing without needing routers or internet connectivity. This setup made the system modular and easily expandable.
- GPS Unit (NEO-6M): To determine the position of both the ambulance and nearby vehicles, we integrated the NEO-6M GPS module. This device continuously output's location data. The receiving vehicle uses this information to calculate how far the ambulance is, and if it comes within a specific distance, a warning is displayed to the driver.
- NRF24L01 Module: This is a compact wireless communication module that operates at 2.4 GHz. We used it to quickly send alerts from the ESP32-CAM to the master controller when an emergency vehicle is detected. Its efficient range and low power use made it ideal for short-distance, real-time communication.
- DFPlayer Mini (Audio Module): This small board allows playback of preloaded audio files from a microSD card. When an emergency vehicle is detected, the DFPlayer is triggered to play alert messages or siren sounds. This creates an audio cue for nearby drivers or pedestrians, improving safety awareness in busy traffic areas.
- Displays (TFT and OLED): Inside vehicles, we installed TFT displays to notify drivers of incoming ambulances. These screens show messages like "Emergency Vehicle Nearby" and update in real-time based on distance. OLED screens were used to monitor GPS status or other values during debugging. Both types of displays supported fast updates and were easy to integrate with our microcontrollers.
- LED Lights and 7-Segment Displays: Each traffic pole featured standard red, yellow, and green LEDs to simulate traffic signals. These were controlled by the slave ESP8266 units. 7-segment displays showed countdowns, letting drivers know how many seconds remained before the signal would change. This gave the system a realistic traffic control interface and improved user experience.

Testing strategies & Test Procedures:

- Microcontroller and Module Integration: The connection between ESP8266 and ESP32-CAM was checked to confirm proper wiring and reliable operation. All essential lines—power, ground, and communication pins—were carefully verified for stability. Basic functionality tests were run to ensure each microcontroller responded as expected to data input and signal logic.
- Power Supply Testing: Each component was tested for proper power delivery. Using a multimeter, voltage and current levels were measured across the ESP boards, GPS modules, LEDs, and wireless transceivers. These readings were monitored during real-time tasks such as communication or LED switching to ensure power drops or spikes did not occur.
- ESP-NOW Communication Test: Test code was uploaded to both the master and slave ESP8266 boards to evaluate peer-to-peer wireless data transfer. The serial monitor was used to confirm that control commands were accurately received. Signal strength and consistency were evaluated across short and medium distances in both open and semi-obstructed spaces.
- Emergency Detection Module: The ESP32-CAM, with a pre-trained machine learning model, was tested for its ability to recognize visual indicators of emergency vehicles, such as flashing light patterns. Sample scenes were presented, and real-time detection performance was observed. Upon successful identification, signal transmission to the main controller was confirmed.
- GPS Module Functionality: The NEO-6M GPS module was tested by collecting live coordinates and displaying them via serial output. Controlled

movement of the module was used to assess whether location updates occurred smoothly. Distance calculations between emergency and nearby vehicles were verified with reference measurements.

- Driver Alert System: Code responsible for calculating proximity and triggering visual alerts was tested on the ESP32 connected to the in-vehicle TFT display. Sample GPS values were used to simulate an approaching ambulance. The screen was monitored to confirm correct messages were shown when the distance fell within the alert threshold.
- Traffic Signal Control Logic: The logic controlling traffic signal LEDs was validated by simulating emergency detection inputs. The master ESP8266 was programmed to send commands to slave boards, which then activated the appropriate LED (green, yellow, or red) based on logic. Each light was timed to remain on for the defined duration, and transitions were checked for accuracy.
- Audio Alert System: The DFPlayer Mini was tested by triggering audio playback from the ESP32-CAM. Pre-recorded sounds were stored on a microSD card, and playback was initiated during detection events. The sound was verified for clarity and proper volume in an outdoor environment to ensure effective driver and pedestrian notification.

VI. CONCLUSION AND FUTURE STUDIES

This project successfully demonstrates an intelligent, autonomous traffic control system designed to prioritize emergency vehicles such as ambulances in urban environments. By integrating edge computing with the ESP32-CAM for real-time visual detection, wireless communication modules (NRF24L01 and ESP-NOW), and GPS-based driver alerts, the system offers a practical and low-latency solution for improving emergency response times. Throughout the development process, challenges such as synchronizing multiple microcontrollers, ensuring reliable wireless communication, and implementing accurate real-time detection were addressed effectively. Extensive testing validated the system's ability to dynamically control traffic signals, provide timely warnings to drivers, and operate independently from cloud infrastructure. The potential applications of this system extend beyond emergency vehicle prioritization, with promising implications for smart city traffic management, road safety enhancement, and intelligent transportation systems. Its modular and decentralized architecture ensures adaptability for various urban settings with minimal infrastructure changes. While the current implementation proves the concept's feasibility and operational effectiveness, several areas remain open for further enhancement. Future work could focus on improving the emergency vehicle detection algorithm through more advanced machine learning techniques to increase accuracy and reduce false positives. Additionally, expanding the system to support multiple emergency vehicles simultaneously and integrating voice or gesture-based alerts for drivers could further enhance safety. Exploring integration with broader smart city frameworks and adding real-time traffic flow analytics may also optimize overall traffic management. Finally, long-term field testing in diverse urban environments will be essential to refine the system's robustness and scalability.

REFERENCES

[1] P. V., R. IV, S. A. A., and C. M. L., "A smart GPS and IoT-based strategy for optimizing ambulance response times," Proceedings of the 2024 2nd International Conference on Advances in Information Technology (ICAIT), Chikkamagaluru, India, 2024.

[2] M. Girish et al., "IoT-driven intelligent traffic management system for emergency vehicles," International Journal of Scientific Research in Science and Technology, vol. 10, no. 3, 2023. DOI: 10.32628/IJSRST52310346.

[3] V. Gupta, V. Verma, Y. Yogita, and M. Sethi, "Emergency response enhancement through intelligent systems," 2022 3rd International Conference on Computing, Analytics and Networks (ICAN), p. 1, 2022. DOI: 10.1109/ICAN56228.2022.10007238.

[4] S. Deshmukh, S. Shaikh, K. Sharma, D. Priya, and V. Kurtadikar, "A review of ambulance tracking technologies," International Journal of Innovative Science and Research Technology (IJISRT), vol. 6, no. 5, pp. 1235–1237, May 2021.

[5] M. Abdeen, M. H. Ahmed, H. Seliem, M. El Nainay, and T. R. Sheltami, "Leveraging smart

healthcare systems to improve ambulance service performance," in Workshop on AI, IoT, and Digital Health at IEEE/ACM CHASE, Washington, D.C., USA, Dec. 2021, pp. 205–209.

[6] Y. Kogan, Y. Levy, and R. A. Milito, "Distributed queue strategies: Comparing FIFO and MED," Telecommunication Systems, vol. 7, nos. 1–3, pp. 299–312, June 1997.

[7] N. Fearn, "How smart ambulance connectivity could transform emergency care and ease hospital pressure," Forbes, Nov. 2019. [Online]. Available: https://www.forbes.com

[8] R. P. Gonzalez, G. R. Cummings, H. A. Phelan, M. S. Mulekar, and C. B. Rodning, "Do longer EMS response times increase rural crash fatalities? A statewide analysis," American Journal of Surgery, vol. 197, no. 1, pp. 30–34, Jan. 2009.

[9] A. Ashour, P. Cameron, S. Bernard, M. Fitzgerald, K. Smith, and T. Walker, "Impact of bystander first aid on trauma survival rates," Emergency Medicine Australasia, vol. 19, no. 2, pp. 163–168, Apr. 2007.

[10] Y.Zhang,"Optimizing Road route selection through integrated traffic planning and design principles," Proceedings of the 2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs), Nicosia, Cyprus.