Smart City Solution: Emergency-Aware and Eco-Friendly Traffic Signal System

Aarathi Dinesh[1], Anusree R[2], Devananda K[3], Swathy Chandran N[4], Jithendra P R Nayak[5]
[1,2,3,4]*Students, Department of Computer Science and Engineering, Srinivas Institute of Technology (S.I.T), Mangalore, Karnataka, India*
[5]*Assistant Professor, Department of Computer Science and Engineering, Srinivas Institute of Technology (S.I.T), Mangalore, Karnataka, India*
doi.org/10.64643/IJIRTV12I12-206769-459

***Abstract*—Urban congestion and deteriorating air quality rank among the most persistent problems modern cities face challenges that have grown more acute as populations expand and vehicle ownership climbs. At the heart of the issue lies the conventional traffic signal, whose operation is governed entirely by pre-set timing cycles with no capacity to sense or react to real conditions at an intersection. The practical consequences are twofold. Emergency vehicles ambulances carrying critical patients, fire engines responding to active incidents have no guaranteed right of way and may be held at red lights just like ordinary traffic. Separately, engines idling at stalled intersections discharge pollutants continuously, representing an ongoing health burden for residents in the vicinity. This paper presents a unified system designed to address both concerns within a single embedded platform. The proposed Emergency-Aware and Eco-Friendly Traffic Signal System pairs two ESP32 microcontrollers: one dedicated to visual detection of emergency vehicles through a YOLOv8-powered pipeline fed by an overhead webcam, and another tasked with continuous air quality measurement via an MQ-135 gas sensor. When an emergency vehicle is recognised with sufficient classifier confidence, that lane receives an immediate green signal while all remaining lanes are held at red. In parallel, the pollution monitor tracks CO_2 and NH_3 concentrations, activating a red warning LED and pushing data to Firebase whenever readings exceed a defined threshold. Concurrent operation of both subsystems was verified across multiple test scenarios, with neither function degrading the other. Future iterations will incorporate cloud-based analytics, GPS-informed pre-clearance, and cross-intersection signal coordination.**

***Index Terms*—ESP32 Microcontroller, Emergency Vehicle Detection, Smart Traffic Management, Air Quality Monitoring, MQ-135, Image Processing, IoT, Smart City.**

I. INTRODUCTION

Managing traffic in large, densely settled cities has never been straightforward, and the challenge deepens every year as urban populations grow and more vehicles compete for the same road space. The fixed-cycle signal systems that currently govern most intersections were built around a simple premise rotate through green, amber, and red on a timer and that premise has not changed meaningfully in decades. Nothing in a conventional signal reads how long the queue is, detects whether an ambulance is approaching, or adjusts to the actual pattern of demand in front of it. The clock cycles regardless.

Two serious consequences follow from this inflexibility. The first concerns emergency response. A paramedic team racing to a cardiac call, or a fire crew heading to a structure fire, cannot count on receiving any assistance from the signal infrastructure they encounter along the way. They must either wait for the light to change or depend on other drivers choosing to move aside, neither of which is reliable. Minutes matter in both scenarios, and the road network currently offers no mechanism for shaving them off. The second consequence is environmental. Every vehicle sitting idle at a red light keeps its engine running, and running engines emit carbon dioxide, nitrogen oxides, and other combustion by-products. For households situated close to a busy junction, that emission is not an occasional event it is a routine part of every day.

The work described here treats both of these problems as a joint design challenge rather than separate issues requiring separate solutions. By integrating real-time computer vision for emergency vehicle recognition

with continuous electrochemical air quality monitoring, the platform can react to what is actually happening at the intersection. Two ESP32 microcontrollers provide the embedded control layer; a YOLOv8-based detection pipeline handles vehicle classification from a live webcam feed; and an MQ-135 sensor reports pollutant concentrations to Firebase at near-real-time cadence. The overall result is a compact, affordable system that slots readily into the broader framework of smart city infrastructure.

II. PROPOSED SYSTEM

The central design objective was to handle two operationally distinct challenges getting emergency vehicles through intersections quickly and monitoring roadside air quality from a single piece of hardware rather than deploying two independent systems side by side. The architecture achieves this by running both functions concurrently on ESP32-based controllers, keeping the combined hardware footprint compact and the per-unit cost low while maintaining dependable real-time performance across both tasks.

Traffic management is handled on the visual side of the platform. A webcam with a clear overhead view of all four approach lanes feeds a continuous video stream to a host PC, which runs a YOLOv8 classification model against each incoming frame. When the model identifies a truck or bus type in any lane with a confidence value that meets the override threshold, it issues a priority command to the signal-control ESP32. That lane transitions immediately to green; every other lane is held at red and kept there until the system confirms the vehicle has passed through. Once the intersection is clear, signal cycling returns to its normal sequence automatically, without any operator action.

The pollution monitoring function runs on a separate ESP32 and operates independently of the visual pipeline. An MQ-135 sensor positioned near the road surface where exhaust concentrations are highest continuously measures the ambient mix of carbon dioxide, nitrogen oxides, and related gases. The microcontroller evaluates each reading against a calibrated safety threshold and drives one of two LED indicators accordingly: green for acceptable air quality, red when the threshold is exceeded. Alert events and status readings are written to a Firebase cloud database at 100-millisecond intervals, making the data accessible to remote monitoring without any on-site presence.

III. METHODOLOGY

A. Hardware Architecture

The physical build centres on two ESP32 development boards with clearly separated responsibilities. The first controls the traffic signals: it receives override commands from the PC-hosted detection pipeline and switches the red, amber, and green LEDs representing each of the four intersection lanes. The second board owns the pollution function: it reads the MQ-135 sensor output, drives the pollution-status LED indicators, and manages the Firebase write cycle. Both boards draw from a stable regulated supply to guard against resets or dropped data during extended operation.

Camera placement follows a single constraint: all four lanes must fall cleanly within the field of view. A 1080p USB webcam mounted overhead achieves this, and the consistent geometry it provides simplifies the lane-boundary assignments made later in the detection pipeline. The MQ-135 sensor is positioned at road level, within the band of highest exhaust concentration, to maximise the sensitivity of pollution readings.

B. Vehicle Detection Pipeline

Each video frame is resized to 640×480 pixels before being passed to the YOLOv8n model, which returns bounding boxes and class labels for every object it detects. Lane assignment works by dividing the image into four equal horizontal zones calculated from the frame's centre coordinates, then mapping each detection's centroid to the appropriate zone. The recognisable vehicle classes are cars, buses, trucks, and motorcycles. Pedestrian detections are treated separately: any lane containing a detected person is flagged and excluded from priority consideration until the pedestrian clears the zone.

Among the remaining vehicle detections, the pipeline checks whether any truck or bus carries a confidence score above 0.45, which is the threshold defining emergency vehicle classification. A temporal smoothing step across consecutive frames acts as a guard against false positives from ambiguous objects a large van at an unusual angle or a toy truck partially visible at the frame edge, for instance before any override command is sent to the ESP32.

C. Pollution Monitoring Logic

The MQ-135 is an electrochemical sensor with broad sensitivity to the pollutant gases most associated with urban vehicle traffic, including carbon dioxide, ammonia, benzene, and nitrogen oxides. It produces an analog voltage proportional to the total gas concentration in its vicinity; the ESP32's 12-bit ADC digitises that voltage to an integer value between 0 and 4095.

Calibration testing established 150 as the dividing point between acceptable and elevated pollution. When a reading exceeds that value, the firmware sets gas_status to PRESENT, activates the red LED, and writes an alert record to Firebase. When the reading falls at or below 150, gas_status reverts to ABSENT and the green LED is restored. Cloud updates are sent every 100 milliseconds, generating a near-continuous log of air quality conditions at the junction that can be reviewed through the Firebase dashboard from any location.

D. Test Results Summary

The table below summarises the black-box test cases used to verify each operational mode:

Test Case	Expected Output	Result
Normal signal operation	Signals cycle normally	Pass predefined sequence maintained
Emergency vehicle detected by webcam	Signal changes to green for that lane	Pass signal overridden immediately
MQ-135 reads above threshold (≥150)	Red LED activates	Pass alert fired correctly
MQ-135 reads below threshold	Green LED activates	Pass safe indicator confirmed
Pedestrian detected in a lane	Lane blocked from priority selection	Pass lane correctly excluded

Table I. Black-Box Test Case Results

IV. SYSTEM DESIGN

Viewed at the architectural level, the system organises naturally into three functional tiers stacked from physical sensing to observable output. The sensing tier comprises the webcam capturing visual information about lane occupancy and the MQ-135, which samples the surrounding air. The processing tier sits above it: the host PC runs object detection and decides which lane should receive priority, while each ESP32 translates incoming data and decisions into physical switching actions. The output tier is what the environment ultimately sees: the traffic signal LEDs change state, the pollution indicator LEDs report air quality, and the Firebase database accumulates a timestamped record of both.

Data flows through the system in two parallel streams that remain independent of one another throughout. The visual stream travels from camera to PC, through the detection and classification pipeline, and arrives at the signal-control ESP32 as a serialised priority instruction. The sensor stream travels from the MQ-135 through its dedicated ESP32, which simultaneously drives the LED indicators and writes records to the cloud. Keeping these paths on separate microcontrollers was a deliberate choice rooted in resource management: the detection pipeline is computationally demanding, and sharing it with sensor monitoring would mean that processing backlogs could delay pollution alerts. Separation prevents that coupling entirely.

The same separation provides a measure of fault tolerance. If the host PC stalls or the detection pipeline temporarily hangs, the pollution monitor carries on without interruption. Equally, a sensor-side issue leaves visual detection unaffected. Each ESP32 maintains its own communications with the rest of the system and can continue operating through a fault on the other side.

V. RESULTS AND DISCUSSION

Validation was organised around four operational scenarios: idle baseline cycling with no vehicles at the intersection; normal operation with vehicles in multiple lanes simultaneously; emergency vehicle detection and signal override; and pollution monitoring at both acceptable and elevated gas concentrations.

Baseline operation matched expectations straightforwardly. Signals stepped through the preset sequence without interruption, and when vehicles appeared across several lanes at once, the system correctly identified the most-occupied lane and brought it to green first, consistent with the intended priority logic.

Emergency vehicle tests introduced a model fire truck into one of the four lane zones. The YOLOv8 classifier identified it as a truck-class object with confidence scores consistently above 0.45 in every run. The override command reached the signal controller within a single processing cycle, placing the correct lane on green while holding all others at red. Removing the model vehicle from frame caused the system to return to normal cycling automatically, with no manual reset required. Notably, this approach requires no hardware modification to the vehicles themselves no RFID transponders, no infrared emitters which represents a substantial practical advantage over alternatives in the literature.

Pollution monitoring was validated by adjusting aerosol concentrations near the sensor above and below the threshold of 150 analog units. The red LED activated reliably each time readings crossed the boundary, and the Firebase database recorded a PRESENT status within a measured latency that stayed consistently below 200 milliseconds. Returning concentrations to safe levels switched the indicator to green and updated the database to ABSENT. The combination of local LED feedback and sub-200-millisecond cloud updates means a remote operator would have a practically real-time view of junction air quality at all times.

Taken together, the results confirm that both subsystems meet their design objectives and do so simultaneously. No observable interference was detected between the detection pipeline and the sensor monitoring loop during any test scenario, validating the architectural decision to keep them on separate hardware.

VI. CONCLUSION

The work reported here addresses a genuine gap in urban traffic infrastructure. Conventional fixed-cycle signals cannot give emergency vehicles a clear path through an intersection, and they offer no mechanism for tracking the pollution that idling vehicles generate at their approach lanes. The system described in this paper confronts both limitations through a single integrated platform assembled from accessible, off-the-shelf components.

Across all tested scenarios, performance was consistent. Emergency vehicle detection relied solely on computer vision with no on-vehicle hardware requirements, and the signal override completed within one detection cycle. Pollution monitoring delivered near-real-time alerts through both local LEDs and cloud reporting, with cloud latency reliably below 200 milliseconds. The modular architecture means either subsystem can be upgraded or extended independently higher-resolution cameras, broader sensor coverage, or additional intersections without requiring changes to the other.

Several directions for future development present themselves. Upgrading the webcam to a higher-resolution unit would improve detection reliability in low-light or night-time conditions where the YOLOv8 model may struggle with motion blur or underexposed frames. Linking adjacent intersections over the ESP32's built-in Wi-Fi would allow the system to extend a rolling green corridor ahead of an emergency vehicle rather than clearing only the immediate crossing. GPS integration at the dispatch level would go further still, providing advance notification while the vehicle is still hundreds of metres away and allowing pre-emptive lane clearance before it even enters sensor range.

ACKNOWLEDGMENT

The authors sincerely thank their project guide, Dr. Jithendra P R Nayak, Associate Professor in the Department of Computer Science and Engineering at Srinivas Institute of Technology, Mangaluru, for his sustained guidance, encouragement, and patience across all phases of this project. Gratitude is also owed to Prof. Deepthi P D'Souza, Project Coordinator; Prof. Aravind Naik, Head of Department, CSE; and Dr. Shrinivasa Mayya D, Dean, for institutional support and access to the required facilities. The contributions of all teaching and non-teaching staff in the CSE department are genuinely appreciated. The authors also wish to record their gratitude to their families and friends, whose encouragement was a steady presence throughout.

REFERENCES

[1] K. R. Sagar and S. N. Kumar, "Smart Traffic Signal Control System Using IoT," in Proc. IEEE Int. Conf. Smart Transportation Systems, 2022.

[2] S. Sharma and A. Mehta, "Emergency Vehicle Priority System Using RFID Technology," Int. J. Eng. Res. Appl., vol. 11, no. 3, 2021.

[3] R. Patel and V. Shah, "Intelligent Traffic Light Control System Using Arduino and Infrared Sensors," in Proc. IEEE Conf. Smart City Technologies, 2020.
[4] M. Kumar and P. Singh, "IoT-Based Air Pollution Monitoring System Using Gas Sensors," IEEE Access, vol. 9, 2021.
[5] J. Lee and H. Kim, "Adaptive Traffic Control System Using Wireless Sensor Networks," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 4, 2019.
[6] Gupta and R. Verma, "Smart City Traffic Management Using Embedded Systems," Int. J. Smart City Appl., vol. 6, no. 2, 2022.
[7] Y. Zhang and L. Wang, "Gas Sensor-Based Air Quality Monitoring System for Urban Environments," Environ. Monit. Assess., 2020.
[8] P. Kumar and S. Roy, "Intelligent Traffic System with Emergency Vehicle Detection," in Proc. IEEE Smart Transportation Conf., 2023.
[9] Rahman and M. Hossain, "Smart Traffic Management System Using ESP32 and IoT Technology," in Proc. IEEE Int. Conf. Smart Cities, 2023.
[10] S. Das and P. Chatterjee, "IoT-Based Intelligent Traffic Signal and Environmental Monitoring System," Int. J. Adv. Embedded Syst., vol. 8, no. 1, 2022.
[11] T. Nguyen and J. Park, "Deep Learning-Based Emergency Vehicle Detection Using CNN," IEEE Trans. Intell. Transp. Syst., 2022.
[12] R. Das and S. Banerjee, "Computer Vision-Based Traffic Control Using Real-Time Video Analysis," Int. J. Comput. Vis. Appl., 2021.
[13] J. Redmon et al., "YOLO: Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018.
[14] George and S. Mathew, "Smart Intersection Management System," in Proc. IEEE Smart City Conf., 2023.
[15] D. Roy and S. Ghosh, "Machine Learning-Based Traffic Signal Control Using Historical and Real-Time Data," in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), 2022.