IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

Multi-Featured Smart Vehicle with Safety Monitoring using IoT

M. Ravi[1], Anup Mandal[2], G. Anirudh[3], H. Bhagya Shree[4], M. Shashank Yadav[5]

Associate Professor, Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, India[1]

Student, Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, India[2,3,4,5]

Abstract: Road traffic accidents remain one of the leading causes of preventable fatalities worldwide, primarily due to delayed hazard perception and delayed emergency response. This paper presents a Multi-Featured Smart Vehicle with Safety Monitoring using IoT, a low-cost, retrofit-capable embedded system that combines real-time obstacle avoidance, blind-spot monitoring, automatic accident detection and GPS-based emergency alerting on a single dual-core ESP32 platform. The system fuses data from an HC-SR04 ultrasonic sensor, two infrared proximity sensors, an MPU6050 accelerometer-gyroscope and a Neo-6M GPS module, and uses an RTOS-based dual-core scheduling strategy to guarantee that collision-avoidance logic is never delayed by network activity. On detecting a severe impact, the system autonomously halts the drive motors and broadcasts an SOS message containing a live Google-Maps location link over Bluetooth to a connected terminal, eliminating dependence on human intervention during the critical "golden hour" after a crash. Experimental evaluation of the prototype across repeated test trials shows a crash-detection accuracy of 96%, a forward-obstacle avoidance efficiency of 95%, a blind-spot alert reliability of 98%, and an average end-to-end SOS response time of 1.45 seconds. These results demonstrate that high-end vehicular safety features, traditionally restricted to expensive luxury vehicles, can be replicated at a fraction of the cost using accessible embedded hardware, making the proposed architecture suitable for retrofitting into economy and public-transport vehicles.

Keywords*:* Internet of Things (IoT), ESP32 Microcontroller, Accident Detection, Obstacle Avoidance, Blind-Spot Monitoring, GPS Tracking, Smart Vehicle Safety, RTOS.

I. INTRODUCTION

The modern automotive landscape is undergoing a profound shift from passive to active safety engineering. For decades, vehicle safety relied on mechanical measures such as seatbelts, reinforced chassis and airbags, all of which mitigate injury only after a collision has occurred. With the exponential growth of vehicle density on public roads, such reactive measures are no longer sufficient. This work introduces a Multi-Featured Smart Vehicle architecture that acts as a proactive safety layer between the driver, the vehicle and the surrounding road environment by combining embedded sensing, real-time decision logic and IoT-based emergency communication.

Despite continual advances in vehicle manufacturing, road accidents remain a critical global public-health problem. Traditional vehicles depend entirely on the driver's perception and reaction time, which is easily compromised by fatigue, distraction, poor visibility or sudden health emergencies, and they lack any autonomous hazard-detection capability. A second, often more fatal, problem lies in post-accident response: victims in remote or unmonitored areas are frequently unable to call for help, and the loss of the trauma-medicine "golden hour" following a crash directly increases the likelihood of a fatal outcome. Commercial Advanced Driver Assistance Systems (ADAS) that address these issues, such as radar-based collision avoidance and autonomous emergency braking, remain complex, proprietary and restricted almost exclusively to luxury vehicles, leaving the majority of the global vehicle fleet without access to these life-saving features.

The motivation for this work is therefore the democratization of vehicular safety technology. Recent miniaturization and cost reduction of components such as the ESP32 microcontroller, MPU6050 accelerometer and Neo-6M GPS module make it feasible to reproduce high-end safety functionality at a fraction of commercial cost, in line with global road-safety initiatives such as Vision Zero, which call for safety systems that can be retrofitted into the millions of vehicles already in use rather than limited to new purchases.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

The specific objectives of this work are to: (i) implement multi-directional, real-time obstacle and blind-spot detection using ultrasonic and infrared sensors; (ii) develop a mathematically filtered impact-sensing mechanism using an accelerometer-gyroscope module capable of distinguishing genuine crashes from routine road vibration; (iii) integrate continuous GPS-based location tracking so that precise coordinates are available for every emergency alert; (iv) exploit the dual-core RTOS capability of the ESP32 to guarantee that collision-avoidance logic executes with minimal latency irrespective of network activity; and (v) design the overall architecture to be cost-effective and scalable for eventual integration with cloud-based fleet-management platforms.

II. LITERATURE SURVEY

The convergence of the Internet of Things, embedded microcontrollers and machine learning has driven significant research into intelligent transportation systems. Alvi et al. [1] conducted a comprehensive study of IoT-based accident-detection systems and emphasized that delayed communication during the golden hour exacerbates traffic fatalities, underscoring the need for continuous vehicular telemetry. Raffik et al. [6] developed an intelligent accident-detection framework based on mechanical sensing to reduce reliance on human bystanders for emergency dispatch, while Josephinshermila et al. [5] proposed an automotive smart black-box using IoT and Zigbee communication for diagnostic logging; both approaches, however, remain largely reactive and lack integrated mechanical actuation.

A parallel body of work explores vision- and learning-based collision detection. Adewopo and Elsayed [4] introduced a deep-learning ensemble using an I3D-CONVLSTM2D architecture over RGB frames and optical flow for smart-city surveillance, and Tian et al. [2] investigated cooperative vehicle-infrastructure systems combined with machine vision to shorten response time. Daisy et al. [7] similarly combined computer vision and machine learning for predictive accident modelling. While these approaches achieve high detection accuracy, they require substantial computational power, camera hardware and network bandwidth, making them impractical for low-cost embedded deployment.

For spatial awareness and tracking, Ikram et al. [3] demonstrated the effectiveness of low-cost IoT proximity sensors for obstacle mapping, and Teja et al. [8] validated the use of lightweight microcontrollers with GPS and cloud dashboards for continuous vehicle tracking, though this system lacked any collision-avoidance mechanism and depended heavily on local wireless coverage. Table I summarizes the reviewed literature.

TABLE I: SUMMARY OF LITERATURE SURVEY

Author (Year)	Methodology / Focus	Key Findings	Limitations Identified
Alvi et al. (2020)	IoT-based accident telemetry and communication networks	Delayed communication during the golden hour worsens fatalities	Focuses on communication; lacks mechanical collision prevention
Tian et al. (2019)	Cooperative Vehicle Infrastructure Systems (CVIS) with machine vision	Improved rescue efficiency using smart-city infrastructure	Highly dependent on external, city-wide infrastructure
Adewopo & Elsayed (2024)	I3D-CONVLSTM2D deep-learning ensemble on RGB/optical-flow frames	Accurate multi-angle collision detection from urban video	Needs immense computation; unsuitable for embedded MCUs
Josephinshermila et al. (2023)	IoT/Zigbee-based automotive smart black-box	Robust post-accident data logging and alert framework	Operates passively; cannot intervene to stop the vehicle
Raffik et al. (2021)	Hardware-based crash sensing and automated notification	Decreased casualty rates via instant authority notification	Limited predictive capability prior to physical impact
Daisy et al. (2025)	Machine learning, computer vision and IoT modelling	Minimized mishaps via environmental and fatigue analysis	Expensive high-end camera and processing requirements
Ikram et al. (2025)	Proximity IoT sensors for environmental mapping	High reliability and low latency of acoustic/IR sensing	Designed for pedestrian use; needs adaptation for vehicles

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

Author (Year)	Methodology / Focus	Key Findings	Limitations Identified
Teja et al. (2023)	NodeMCU, GPS and ThingSpeak cloud tracking	Effective continuous location tracking and theft prevention	Reliant on Wi-Fi/cellular range; no collision avoidance

The reviewed literature reveals three consistent gaps: (i) high computational overhead in deep-learning and vision-based methods that prevents deployment on low-cost microcontrollers; (ii) a lack of integrated mechanical actuation in many IoT monitoring solutions, which detect hazards but cannot physically intervene; and (iii) heavy dependence on external network infrastructure, which is unreliable in rural or unmonitored areas. The proposed Multi-Featured Smart Vehicle System addresses these gaps by shifting from computationally heavy image processing to high-speed, localized sensor fusion on a dual-core ESP32, coupled with autonomous braking actuation and GPS-encoded SOS alerts that operate independently of complex external infrastructure.

III. PROPOSED METHODOLOGY

The proposed system operates as a continuous, real-time IoT cycle rather than a system that merely displays static dashboard information. The operational cycle is divided into five stages: (1) Input and Data Collection, in which forward ultrasonic and side infrared sensors, an accelerometer-gyroscope and a GPS module continuously acquire spatial, kinetic and location data; (2) Pre-processing, where mathematical filtering removes false-positive readings such as small fast-moving debris; (3) Model and System Development, in which the ESP32 executes a concurrent RTOS loop that prioritizes collision-avoidance tasks over non-critical network updates; (4) Algorithm Execution, where filtered sensor data are compared against pre-defined safety thresholds to judge obstacle proximity, unsafe speed or crash occurrence; and (5) Output Generation, in which the system triggers buzzers, actuates braking, or broadcasts an SOS alert with live GPS coordinates depending on the severity of the detected hazard.

A. Hardware Components

The hardware suite is chosen to balance computational capability with cost-effective scalability. The ESP32 Development Board forms the central processing unit; its dual-core processor and native Wi-Fi/Bluetooth capability enable continuous real-time priority loops together with IoT connectivity. Forward obstacle sensing is handled by an HC-SR04 ultrasonic sensor, which measures the time-of-flight of a reflected sound pulse to calculate frontal distance. Left and right Infrared (IR) proximity sensors provide binary blind-spot detection with near-zero processing overhead. An MPU6050 accelerometer-gyroscope module continuously measures 3-axis G-force and tilt to distinguish genuine crashes from ordinary road vibration, while a Neo-6M GPS module supplies live latitude-longitude coordinates for every emergency alert. Physical actuation is performed through an L298N motor-driver bridging the ESP32 to the drive DC motors, complemented by an active piezo buzzer for immediate acoustic warnings. The complete assembly is powered by a rechargeable 18650 Li-ion battery pack capable of sustaining both the logic circuitry and the current-hungry motor and communication peripherals simultaneously.

TABLE III: KEY HARDWARE COMPONENTS AND THEIR FUNCTION

Component	Specification	Function in System
ESP32 Development Board	Dual-core, Wi-Fi/Bluetooth MCU	Central processing and IoT connectivity
HC-SR04 Ultrasonic Sensor	2 cm – 400 cm range	Forward obstacle distance measurement
IR Proximity Sensor (×2)	Digital HIGH/LOW output	Left/right blind-spot intrusion detection
MPU6050 Accelerometer/Gyroscope	3-axis, I2C interface	Impact (G-force) and rollover detection
Neo-6M GPS Module	UART, 9600 bps, NMEA output	Live latitude/longitude telemetry
L298N Motor Driver	H-bridge, dual channel	Motor direction/speed and emergency braking
Active Piezo Buzzer	Digital GPIO trigger	Acoustic driver warning
18650 Li-ion Battery Pack	2 × 3.7 V (7.4 V total)	Stable power for logic and actuation

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

B. Software Development Environment

The embedded firmware was developed in the Arduino IDE (v2.x) using Embedded C/C++ for low overhead and deterministic execution. Sensor interfacing relies on the TinyGPS++ library for parsing NMEA coordinate strings, the Adafruit_MPU6050 and Wire libraries for I2C-based accelerometer communication, and the ESP32Servo library for jitter-free PWM signal generation. The digital output and emergency-notification layer integrates with a Bluetooth Terminal application acting as the primary IoT node, forwarding live telemetry and SOS strings, including embedded Google Maps location links, to registered emergency contacts or a centralized monitoring dashboard.

C. System Architecture

The overall architecture follows a modular design comprising an Input Unit, a Processing Unit, an Output Unit and a Communication Unit, all powered by a common battery supply, as shown in Fig. 1. The Input Unit aggregates ultrasonic, infrared, kinetic and GPS data and forwards it to the ESP32 Processing Unit, which applies threshold-based decision logic to determine whether to issue a warning, halt the vehicle, or initiate an emergency broadcast. The Output Unit, comprising the L298N driver, DC motors and piezo buzzer, executes the resulting physical response, while the Communication Unit uses the ESP32's built-in Bluetooth interface to transmit GPS-tagged SOS alerts to a paired smartphone terminal whenever an accident is confirmed.

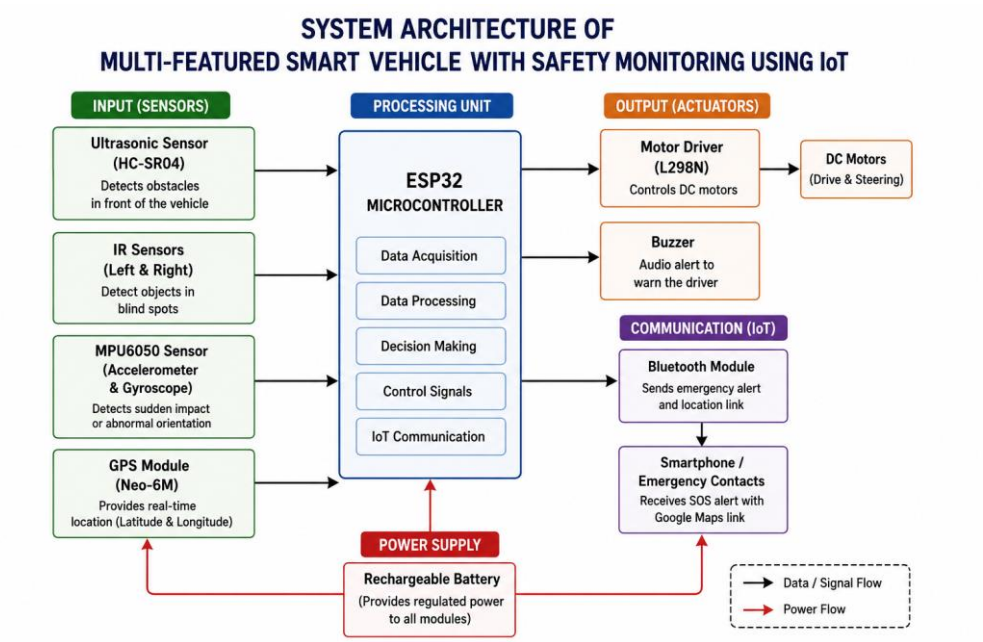

Fig. 1 Overall system architecture of the proposed smart vehicle.

IV. SYSTEM IMPLEMENTATION

Implementation focuses on translating the architectural design into a fully functional prototype, addressing power isolation, sensor-protocol interfacing and deterministic firmware execution so that sensing, processing and actuation operate in synchronization without hazardous latency.

A. Hardware Integration and Circuit Design

The ESP32 acts as the central intelligence hub, with the circuit designed to isolate the electrical noise generated by the drive motors from the sensitive logic components. The complete wiring, connecting the ESP32, sensor array and L298N motor driver, is shown in Fig. 2. The vehicle is powered by a series-connected 2×18650 Li-ion pack delivering approximately 7.4 V; a DC-DC buck converter regulates this to the 3.3 V/5 V logic levels required by the microcontroller and sensors, while the unregulated 7.4 V line is routed directly to the L298N to supply the high-current demand of the drive motors.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

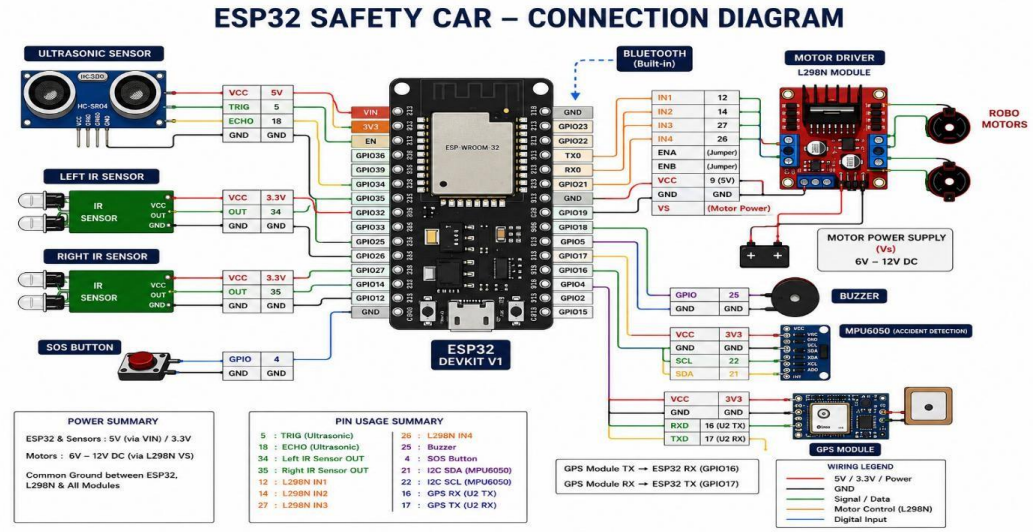

Fig. 2 Complete circuit diagram interconnecting the ESP32, sensor array and L298N motor driver.

B. Sensor Interfacing and Communication Protocols

The HC-SR04 trigger/echo pins are connected to dedicated GPIOs, with frontal distance computed from a 10-µs trigger pulse and the corresponding echo travel time. The IR sensors output a simple binary HIGH/LOW proximity signal, requiring negligible processing overhead. The MPU6050 communicates over the I2C bus (SDA/SCL) to continuously stream 3-axis orientation data, and the Neo-6M GPS module streams NMEA sentences over UART at 9600 bps. On the output side, the L298N is driven through an H-bridge logic interface (IN1–IN4) with PWM-enabled ENA/ENB pins for speed control, and the active piezo buzzer is switched directly by a digital GPIO pin.

C. Embedded Software and RTOS Scheduling

The firmware leverages the ESP32's dual-core architecture through FreeRTOS, pinning Bluetooth/network communication to Core 0 and time-critical sensor polling and motor control to Core 1, so that a dropped wireless connection can never stall the collision-avoidance loop. At start-up, the system performs sensor calibration, sampling the stationary MPU6050 to establish a gravity/tilt baseline. During operation, the main loop continuously evaluates the G-force vector magnitude against a crash threshold (empirically set at approximately 3.5 g); if exceeded, the ESP32 halts the drive motors, retrieves the latest GPS fix and formats an SOS string. In parallel, the ultrasonic distance is compared against a safe-following threshold (approximately 30 cm), overriding user motor commands to prevent frontal collisions.

The core decision logic can be summarized by two threshold conditions evaluated on every iteration of the Core-1 loop:

$$\text{Crash condition: } |G| = \sqrt{(Gx^2 + Gy^2 + Gz^2)} > G\_th \quad (G\_th \approx 3.5\ g) \qquad ...(1)$$

$$\text{Obstacle condition: } D\_front < D\_safe \quad (D\_safe \approx 30\ cm) \qquad ...(2)$$

where Gx, Gy and Gz are the calibrated accelerometer readings along the three axes and D_front is the filtered ultrasonic distance reading. Satisfying equation (1) triggers the SOS/braking protocol, while satisfying equation (2) triggers automated deceleration to prevent a frontal collision; both conditions are checked independently and can pre-empt any concurrent user motor command.

D. IoT Alert Mechanism

When a crash is confirmed, the ESP32 formats a text string combining an emergency message with the live GPS coordinates into a clickable Google Maps URL and transmits it over Bluetooth Serial Port Profile (SPP) to a paired smartphone terminal. This simulates an automated SMS/cloud alert, ensuring that emergency contacts, hospitals or fleet managers receive an accurate location pin at the exact moment an accident occurs, without requiring manual intervention from the victim.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

V. RESULTS AND DISCUSSION

The assembled prototype was subjected to repeated controlled trials to evaluate the reliability, accuracy and latency of each safety subsystem. Fig. 3 shows representative Bluetooth terminal output screens generated by the system, including the accident SOS alert, manual SOS-button alert and obstacle-detection notifications.

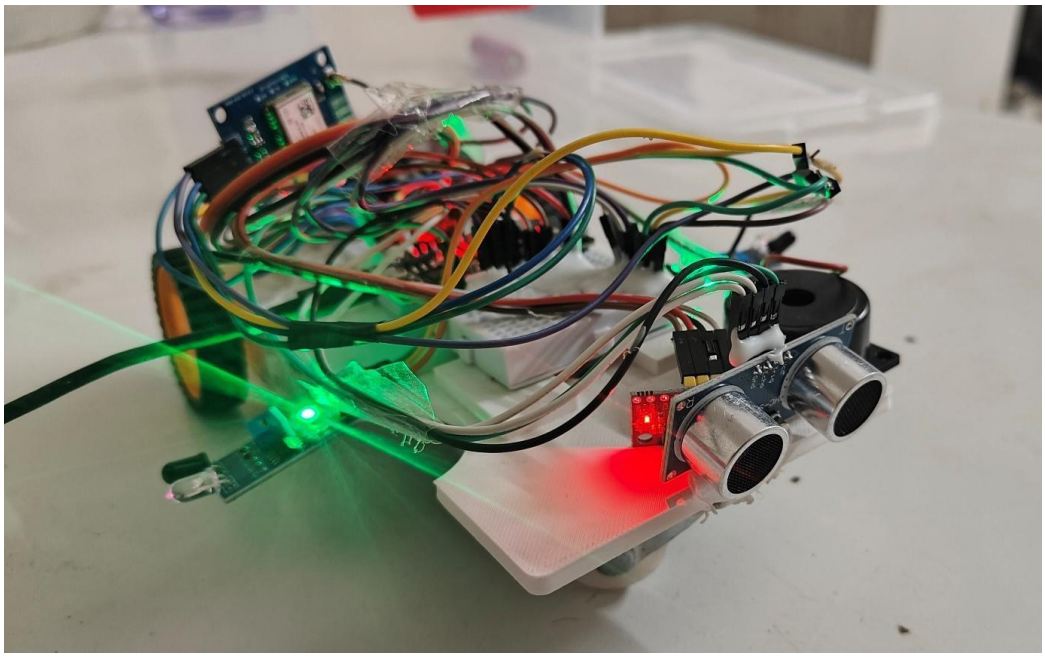

Fig. 3 Bluetooth terminal output screens showing accident alert, SOS-button alert and obstacle-detection notifications.

A. Performance Evaluation

The crash-detection algorithm was evaluated on its ability to distinguish genuine impacts from ordinary road vibration; across 50 simulated impact tests, 48 correctly triggered the SOS protocol, giving a crash-detection accuracy of 96%. The obstacle-avoidance system was evaluated over 100 obstruction encounters, of which 95 resulted in a successful automated halt (95% efficiency), with minor reductions in performance observed against highly sound-absorptive surfaces. The blind-spot monitoring system achieved a 98% alert reliability across 50 lane-change test cases, consistently triggering the buzzer within 15 ms of an object entering the 15 cm threshold. The end-to-end SOS response time, comprising MPU6050 interrupt processing (≈50 ms), GPS NMEA parsing (≈200 ms) and Bluetooth transmission (≈1.2 s), averaged 1.45 seconds, confirming near-instantaneous emergency notification. These results are summarized in Table II.

TABLE II: SUMMARY OF PERFORMANCE EVALUATION

Performance Metric	Test Basis	Result
Crash Detection Accuracy (MPU6050)	48 / 50 correctly triggered SOS events	96%
Forward Obstacle Avoidance Efficiency	95 / 100 successful automated halts	95%
Blind-Spot Monitoring Reliability	49 / 50 correct lane-change alerts	98%
Average End-to-End SOS Response Time	Interrupt + GPS parse + BT transmit	1.45 s

B. Latency Analysis

Fig. 4 presents the breakdown of functional latency across the system's principal response stages. The digital braking response (≈0.10 s) reflects the mechanical activation time between an unsafe ultrasonic reading and motor power cut-off, while internal processing latency (≈0.05 s) illustrates the negligible computational delay afforded by the ESP32's 240 MHz dual-core processor. The IoT handshake and dispatch stage (≈1.20 s), which formats GPS coordinates into a URL

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15714

and broadcasts it over the wireless serial link, is the dominant contributor to total latency; even so, a 1.2-second dispatch delay is well within acceptable limits for real-world automotive emergency-response standards.

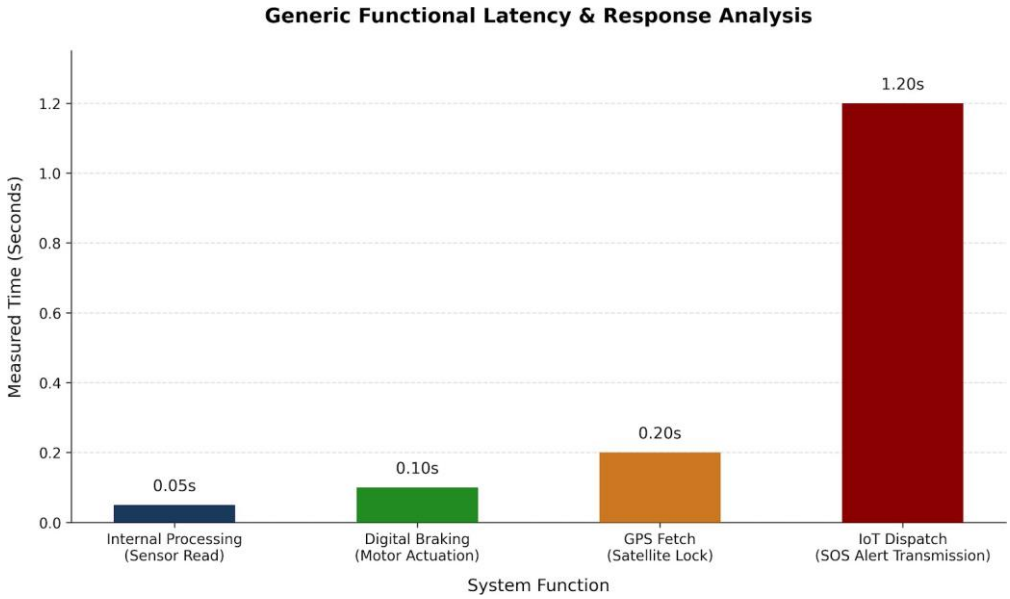

Fig. 4 Functional latency and response-time analysis of the safety subsystems.

Overall, the results validate the central design hypothesis: by replacing computationally expensive vision-based detection with localized, threshold-based sensor fusion on a dual-core microcontroller, the proposed system achieves reliable, low-latency accident prevention and emergency notification at a fraction of the cost of proprietary ADAS solutions, while remaining lightweight enough to be retrofitted into existing economy vehicles.

C. Comparison with Existing Approaches

Unlike the vision- and deep-learning-based systems surveyed in Section II [2], [4], [7], which demand high-bandwidth video streaming and substantial onboard computation, the proposed system performs all safety-critical decision-making using lightweight threshold logic on a single microcontroller, making it viable for low-cost economy vehicles rather than being restricted to research or luxury platforms. In contrast to passive black-box and tracking solutions [3], [5], [8], which log data or report location but cannot physically intervene, the proposed architecture couples detection directly to actuation, autonomously halting the vehicle and issuing an SOS alert without requiring any external network infrastructure beyond a paired Bluetooth terminal. This combination of low computational cost, autonomous actuation and infrastructure-independent emergency communication distinguishes the proposed work from the systems reviewed in the literature.

VI. CONCLUSION AND FUTURE SCOPE

This paper presented the design, implementation and evaluation of a Multi-Featured Smart Vehicle with Safety Monitoring using IoT, aimed at democratizing advanced vehicular safety through an accessible, cost-effective embedded platform. By exploiting the dual-core RTOS capability of the ESP32 alongside an HC-SR04 ultrasonic sensor, IR proximity sensors, an MPU6050 kinetic module and a Neo-6M GPS tracker, the prototype achieved rapid-response spatial awareness, automated collision mitigation and autonomous GPS-encoded emergency notification, addressing one of the most fatal flaws in modern transportation, delayed emergency response. Experimental evaluation demonstrated a 96% crash-detection accuracy, 95% obstacle-avoidance efficiency, 98% blind-spot alert reliability and an average SOS response time of 1.45 seconds, confirming the viability of the proposed architecture for real-world deployment.

The current prototype has been deliberately designed to be scalable, allowing several advanced extensions in future iterations:

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15714

1) Cloud-Based Fleet Management: upgrading the IoT communication suite from a localized Bluetooth terminal to a centralized cloud dashboard (e.g., AWS IoT or Firebase) so that transport operators can simultaneously track the location, speed and safety status of large vehicle fleets in real time.

2) Edge Artificial Intelligence: replacing the current fixed mathematical thresholds with on-device Ensemble Machine Learning models, such as Decision Trees, Support Vector Machines, K-Nearest Neighbors, Random Forest and Gaussian Naive Bayes, enabling the system to learn driver-specific behavior and predict collision probability dynamically rather than react to fixed limits.

3) Computer Vision Integration: augmenting the ultrasonic/IR sensor suite with a camera module and an OpenCV-based microcomputer (e.g., Raspberry Pi) to enable lane-departure warnings, traffic-sign recognition and pedestrian tracking.

4) Automated Medical Dispatch: interfacing the SOS protocol directly with local emergency-dispatch APIs so that ambulance services receive crash-severity data derived from G-force measurements automatically, rather than only a location link sent to a pre-programmed contact.

ACKNOWLEDGMENT

The authors thank the Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad, for providing the infrastructure and support necessary to carry out this work.

REFERENCES

[1] U. Alvi, M. A. K. Khattak, B. Shabir, A. W. Malik and S. R. Muhammad, "A Comprehensive Study on IoT Based Accident Detection Systems for Smart Vehicles," IEEE Access, vol. 8, pp. 122480-122497, 2020.
[2] D. Tian, C. Zhang, X. Duan and X. Wang, "An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems," IEEE Access, vol. 7, pp. 127453-127463, 2019.
[3] S. Ikram, I. S. Bajwa, A. Ikram, I. de la Torre Díez, C. E. Uc Ríos and Á. Kuc Castilla, "Obstacle Detection and Warning System for Visually Impaired Using IoT Sensors," IEEE Access, vol. 13, 2025.
[4] V. A. Adewopo and N. Elsayed, "Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection," IEEE Access, vol. 12, 2024.
[5] P. Josephinshermila, S. Sharon Priya, K. Malarvizhi, R. Hegde, S. Gokul Pran and B. Veerasamy, "Accident Detection using Automotive Smart Black-Box based Monitoring System," Measurement: Sensors, vol. 27, 100721, 2023.
[6] R. Raffik, M. M. Jones, T. Murugajothi and B. Kannadasan, "Intelligent Accident Detection and Smart Alert System for Vehicles," International Journal of Mechanical Engineering, vol. 6, no. 3, Dec. 2021.
[7] I. J. Daisy, A. Arun, E. J. Antony and K. B. Muhilan, "Smart Vehicle Assistance and Accident Prevention System," International Research Journal on Advanced Engineering Hub (IRJAEH), vol. 3, no. 3, pp. 412-419, Mar. 2025.
[8] P. S. Teja, N. S. S. Reddy, P. K. Pandugu and P. Vangari, "Smart Vehicle Monitoring and Tracking System," E3S Web of Conferences (ICMED-ICMPC), vol. 391, 01099, 2023.
[9] N. Lu, N. Cheng, N. Zhang, X. Shen and J. W. Mark, "Connected Vehicles: Solutions and Challenges," IEEE Internet of Things Journal, vol. 1, no. 4, pp. 289-299, Aug. 2014.
[10] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi and M. A. Hossain, "A Survey on Sensor-Cloud: Architecture, Applications, and Approaches," International Journal of Distributed Sensor Networks, vol. 9, no. 2, 2013.
[11] H. Hartenstein and K. P. Laberteaux, VANET: Vehicular Applications and Inter-Networking Technologies, John Wiley & Sons, 2010.
[12] Espressif Systems, ESP32-WROOM-32 Datasheet, Version 3.9, Shanghai, China, 2023.
[13] TDK InvenSense, MPU-6050 Six-Axis Motion Tracking Device Product Specification, Revision 3.4, 2023.

