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. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412150

IoT-Based Railway Track Fault, Obstacle, and Fire Detection Robot

Kavya BS[1], Mr. Yadhu Krishna M R [2], Harshitha A[3], Meghamala N[4], Mohammed Luqmaan[5]

Department of Information Science and Engineering, The Oxford College of Engineering,

Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[1-5]

Abstract: Railway safety continues to be a critical challenge in modern transportation systems due to recurring issues such as track defects, wildlife interference, and fire-related incidents, which frequently lead to severe accidents. This paper presents an intelligent IoT-enabled autonomous robotic system designed to address these challenges through a multi-modal hazard detection approach. The proposed robot autonomously navigates railway tracks while capturing visual data for structural assessment using edge detection techniques and Hough transformation to identify cracks and defects. In parallel, the system monitors wildlife intrusion and fire hazards to prevent potential collisions and emergencies. Detected information is transmitted wirelessly via IoT infrastructure to control centers, enabling real-time monitoring and rapid response actions. Machine learning models, including YOLO for object recognition and MobileNet-SSD for wildlife detection, are utilized to enhance detection accuracy. Experimental results from field tests demonstrate reliable performance with a low false alarm rate and high detection accuracy, while maintaining a cost-effective implementation of approximately INR 5,000 per unit. By reducing dependence on manual inspections and enabling continuous surveillance, the proposed system significantly enhances railway safety and operational efficiency.

Keywords: Railway safety, Autonomous robot, IoT-based monitoring, Computer vision

I. INTRODUCTION

Every day, trains across India transport millions of passengers and vast amounts of goods, making railways the country's primary mode of transportation. They connect distant regions at low cost and play a vital role in economic and social growth. However, ensuring safety is challenging due to the vast rail network, constant usage, and harsh weather conditions that cause track deterioration. Common causes of train disasters include broken rails, track obstructions, animals on tracks, and fires, often resulting from aging materials, poor maintenance, nearby objects, or overheating and fuel leaks. Even small, unnoticed defects can lead to major accidents, loss of life, damage to equipment, and disruption of services.

Traditionally, railway maintenance relies on staff walking along tracks to inspect for defects, but long stretches often remain unchecked between scheduled patrols. Human limitations such as fatigue, poor visibility due to weather or darkness, and differences in experience can cause hazards to go unnoticed for long periods. This makes the inspection process strenuous, time-consuming, and prone to error, allowing problems to grow before they are detected.

To overcome these challenges, compact automated machines and robots are now being used to monitor railway tracks continuously. Equipped with sensors, cameras, ultrasonic detectors, fire and gas monitors, these systems transmit real-time data wirelessly and detect cracks, obstructions, smoke, or heat changes. Engineers receive instant alerts, enabling faster responses and preventive action. Advances in machine vision and automation have made inspections quieter, faster, more reliable, and cost-effective, significantly reducing the risk of missed faults and improving overall railway safety.

II. PROBLEM STATEMENT AND OBJECTIVE

Around most railways today, people still walk the tracks to inspect them by hand, a slow method that catches only some problems. Scheduled patrols follow fixed routes and often miss sudden changes between visits. In remote areas beyond streetlights, fewer officers pass, increasing the chance that issues slip through unnoticed. Problems can surface after inspections end, and mistakes may pile up later due to distraction, fatigue, or loss of focus. Individual judgment and experience strongly influence what gets noticed. Weather further complicates checks—mist, rain, darkness, fog, and storms blur vision and hide defects. Signals weaken, sensors struggle, and machinery strains under changing conditions. Cracks may form quietly, metal snaps may go unheard, and objects on tracks can halt movement without warning. Trouble beneath moving parts often appears suddenly. A single safeguard is rarely enough, as multiple risks can occur at once. Safety is only certain when every hidden danger is detected in time.
Key objectives of MedGuard Edge:

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2025.1412150

- Detects cracks and defects in railway tracks using an ESP32-CAM with basic vision techniques.
- Identifies animals and obstacles near or on the tracks using object detection and ultrasonic sensors.
- Detects fire, smoke, and harmful gases through flame and gas sensors on the robotic unit.
- Sends hazard information instantly to the control center using IoT-based wireless communication.
- Reduces reliance on manual inspections, improving passenger safety, wildlife protection, and infrastructure reliability.
- Maintains low implementation cost (approximately INR 5,000 per unit), enabling large-scale deployment in resource-limited regions.

III. SCOPE

This project presents the development of a railway safety monitoring system that integrates sensor-based hardware, embedded controllers, and IoT communication to enhance track safety and early hazard detection. The system includes modules for image acquisition, crack identification, obstacle and wildlife detection, fire detection, and alert notification. It employs components such as ESP32-CAM, Arduino/ESP32 controllers, ultrasonic sensors, gas sensors, and flame sensors to sense and process safety-critical conditions.

The system is designed for deployment in high-risk environments, including forest regions with frequent wildlife movement, rural and suburban tracks prone to livestock crossings, dry areas vulnerable to fire hazards, and busy railway routes requiring continuous monitoring. By identifying threats in advance, the project contributes to improved passenger safety, accident prevention, and wildlife conservation while offering a cost-effective solution for resource-limited regions. Future enhancements may focus on improving sensor precision and communication reliability, along with the integration of drone-based surveillance, advanced detection algorithms, cloud-based data analysis, and faster communication technologies.

IV. LITERATURE REVIEW

1. Autonomous Robots for Railway Track Inspection
   Patel et al. [2] reported the use of sensor- and camera-equipped mobile robots for railway track inspection. Their approach reduces human involvement and improves inspection safety, though high implementation costs remain a challenge for large-scale adoption.
2. Multi-Sensor Integration in Railway Safety Systems
   Singh and Kumar [3] demonstrated that integrating data from multiple sensors significantly enhances fault detection accuracy and system reliability compared to single-sensor methods.
3. Ultrasonic Sensor-Based Obstacle Detection
   Chen et al. [4] analyzed ultrasonic obstacle detection techniques and found them effective for short-range sensing, although performance may vary under adverse environmental conditions.
4. Fire Detection in Railway Environments
   Gupta and Verma [5] studied railway fire detection systems and concluded that combining flame and gas sensors improves early detection while minimizing false alarms.
5. NodeMCU ESP8266 in IoT Systems
   Wang et al. [6] highlighted the suitability of the NodeMCU ESP8266 for IoT applications due to its low cost, built-in WiFi support, and capability for real-time monitoring.
6. Arduino-Based Sensor Networks
   Reddy and Rao [7] presented Arduino-based monitoring systems, noting their flexibility, simplicity, and effectiveness in handling multi-sensor data collection.
7. Web-Based IoT Monitoring Platforms
   Desai et al. [8] proposed a web-based monitoring framework using IoT and Flask, enabling real-time visualization and remote system access through dashboards.
8. Motor Control for Mobile Robots
   Brown and Smith [9] emphasized that efficient motor control is essential for stable motion and precise navigation in mobile robotic systems.
9. Wireless Sensor Networks in Railways
   Sharma et al. [10] reviewed wireless sensor networks for railway applications, identifying communication reliability and power efficiency as major challenges.
10. Ultrasonic Sensor Performance Evaluation
    Lee et al. [12] evaluated the performance of HC-SR04 ultrasonic sensors and found them reliable for obstacle detection when properly calibrated, with acceptable accuracy and response time.

DOI: 10.17148/IJARCCE.2025.1412150

4.1 Gaps or Areas for Improvement

Most existing railway safety systems address only a single type of threat, while integrated solutions capable of detecting structural faults, wildlife presence, and fire hazards remain limited. Many advanced systems are costly, making them impractical for regions with constrained resources, highlighting the need for affordable solutions with reliable detection performance. Several available approaches rely on off-site processing, resulting in delays; therefore, real-time hazard response using edge-based processing requires improvement. Current systems often depend on human supervision, indicating the need for fully autonomous inspection robots with self-navigation, hazard detection, and alert mechanisms. System performance is also affected by adverse weather conditions such as fog, rain, and extreme temperatures, emphasizing the importance of environmentally robust designs. Additionally, safety solutions must scale effectively across large railway networks without increasing cost or complexity.

This project addresses these limitations by proposing an integrated, low-cost autonomous robotic system that combines multiple hazard detection techniques with real-time processing and alert generation, making it suitable for large-scale railway network deployment.

V. SYSTEM ARCHITECTURE

A. Architecture Overview:

The proposed system follows a modular architecture that integrates hardware sensors, embedded controllers, wireless communication, and intelligent processing algorithms, as illustrated in Figure 1. The architecture consists of five interconnected layers. The sensor layer includes ESP32-CAM for visual crack detection, dual HC-SR04 ultrasonic sensors for obstacle sensing, MQ-135 for air quality and combustible gas monitoring, MQ-136 for hydrogen sulfide and sulfur dioxide detection, GP2Y1010AU0F for particulate matter measurement, flame sensors for fire detection, and a GPS module for location tracking. The processing layer comprises the ESP32-CAM for on-device vision processing, Arduino Uno for sensor data coordination, NodeMCU (ESP8266) for wireless communication, and edge computing for real-time decision making. The communication layer supports WiFi-based data transfer using HTTP and MQTT protocols with JSON formatting and fallback mechanisms to ensure reliability. The application layer provides cloud-based services such as real-time visualization, alert generation, historical data storage, and predictive maintenance analysis. The interface layer enables user interaction through a web-based dashboard, mobile application for field personnel, administrator control panel, and emergency response integration.

B. Data Flow

The system operates through a structured data flow in which sensors continuously collect environmental data at predefined intervals, followed by local preprocessing to remove noise and validate measurements. Captured images are analyzed using computer vision techniques to identify defects, and machine learning models classify detected objects and potential hazards. The processed information is then combined with GPS coordinates and timestamps and transmitted wirelessly to the cloud infrastructure. The cloud platform further processes incoming data streams, evaluates hazard severity through alert algorithms, and sends notifications to the appropriate personnel, while the monitoring dashboard updates in real time to reflect system status.

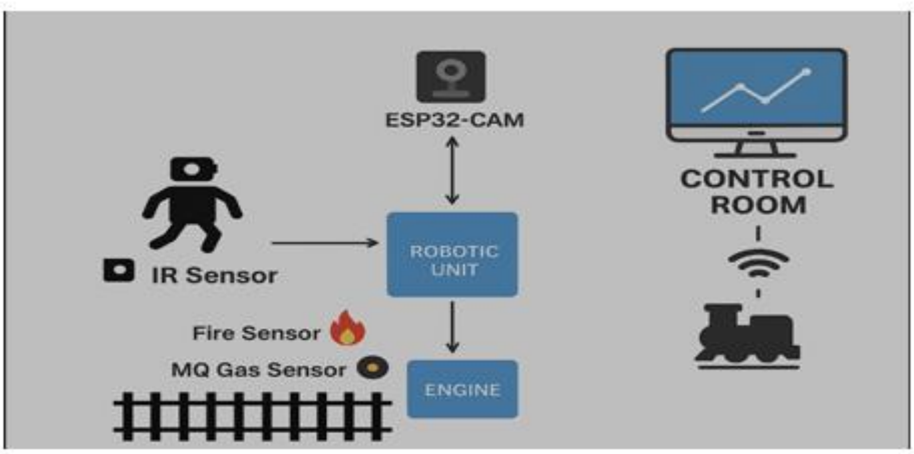

Fig. 1. System Architecture Diagram

© IJARCCE This work is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412150

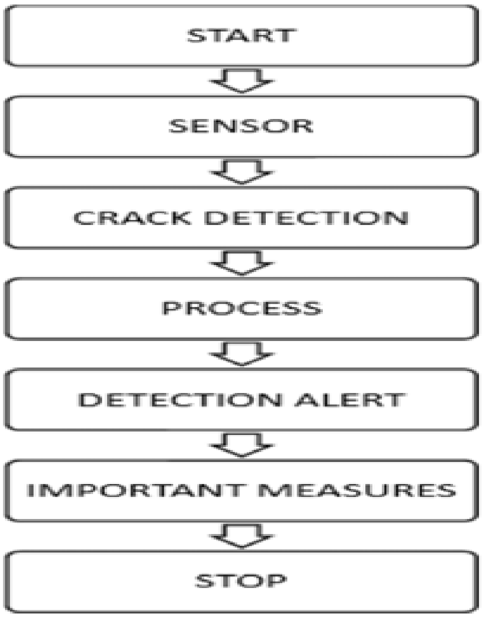

Fig. 2. Data Flow Diagram

VI. METHODOLOGY

The proposed system employs a well-defined methodology that guarantees the trustworthy identification of railroad track faults, obstacles, and fire risks. The methodology incorporates the use of real-time data by means of local processing, secure communication, and efficient monitoring. Though the system is predominantly operated automatically with the least human intervention, it still uses various sensors to consistently monitor the conditions of the railway track.

6.1 IoT Data Gathering

Continuous monitoring of the railway track's surface by crack detection sensors is done, and at the same time, ultrasonic sensors are getting the distance between the robot and any object on the track. Flame sensors are alerting to possible fires by sensing infrared radiation. At the same time, a camera module is taking visuals whenever an unusual situation is detected. The readings of all sensors are gathered in real-time and sent to the microcontroller for further processing.

6.2 Preprocessing of Local Data

In this process, noise is eliminated, invalid readings are removed, and the readings are checked against specified values to identify any unusual situations. The sensor data is standardized and verified to rule out the chances of false alarms due to environmental noise. Local preprocessing is very important in terms of both reducing the amount of data transmitted and increasing the reliability of the detection of hazards.

6.3 Device Cluster Formation

The different parts of the system are organized into one operational cluster logically, which includes the sensors, the NodeMCU ESP8266, the communication modules, and the camera unit. This arrangement allows all the devices in the cluster to work at the same time. The implementation of clustering improves data management, removes communication lag, and assures the concurrent operation of all hardware components.

6.4 Local Decision Processing

The system, rather than dispatching the raw data to the cloud, manages local decision making at the controller's level. The NodeMCU monitors the sensor data that has gone through preprocessing and determines if the condition detected lies within the limits that are considered safe or if it is a danger that could happen.This technique enhances the system's performance not only but also lowers down the network dependency.

6.5 Data Transmission and Cloud Storage

The very moment a dangerous condition is verified, the system sends alert information with sensor values and taken images to the cloud server through Wi-Fi communication. The cloud platform keeps the data with the correct timestamps

DOI: 10.17148/IJARCCE.2025.1412150

for later use. The storage in the cloud enables data analysis of a long period and helps maintenance planning by revealing the areas of the track which are frequently affected.

6.6 Monitoring and Visualization

The data that is stored is made available through a web interface based on Python which was made with Anaconda. The frontend provides alerts in real-time, shows readings from sensors, and gives visual proof of the hazards that have been detected. Such an interface makes it possible for the railway agencies to monitor the conditions of tracks from a distance and also to act very fast in case of emergencies. The historical data can be used too, to back up the inspection and maintenance decisions.

6.7 Alert Generation

The system usually generates alerts immediately whenever a fault or risk is detected and these alerts are displayed on the monitoring interface. The alerts are an aid for the railway staff to take timely preventive actions such as reduced speed for trains, stopping operations, or dispatching maintenance teams. The entire response process contributes to making the situation safer and, consequently, reducing the probability of accidents in the first place.

VII. IMPLEMENATION ENVIRONMENT

The hardware components and specifications of the railway safety robot system designed for reliable and cost-effective operation. The system follows a dual-microcontroller architecture that separates sensor handling and communication tasks, improving robustness, fault isolation, and ease of maintenance. Computer vision–based crack detection is handled independently from sensor acquisition. The Arduino Nano acts as the primary controller responsible for data collection and local control operations. It is built on the ATmega328P microcontroller running at 16 MHz, with 32 KB Flash memory, 2 KB SRAM, and 1 KB EEPROM. The board provides 14 digital I/O pins, including 6 PWM outputs, and 8 analog inputs, operating at 5 V with a 7–12 V input range. Its compact size and Mini-B USB interface make it suitable for embedded deployment. The Arduino performs analog-to-digital conversion for MQ gas sensors, interfaces with ultrasonic sensors using trigger-echo signals, processes flame sensor inputs, controls an LCD via I2C, drives a buzzer for alerts, communicates serially with NodeMCU and ESP32-CAM modules, and sends control signals to the L298N motor driver.

Wireless communication is handled by the NodeMCU ESP8266, which features a 32-bit Tensilica L106 processor operating at 80/160 MHz, 128 KB RAM, and 4 MB Flash memory. It supports 2.4 GHz Wi-Fi connectivity and operates at 3.3 V using an onboard regulator. The module manages network connectivity, implements HTTP and MQTT protocols, formats data in JSON, transmits sensor readings to the cloud, supports OTA updates, and ensures reconnection during network failures. Vision-based inspection is carried out by the ESP32-CAM module equipped with a dual-core LX6 processor at 240 MHz, OV2640 camera, 520 KB SRAM, 4 MB PSRAM, and onboard Flash. It captures high-resolution images, performs preprocessing, transmits data to the server, stores images with timestamps, and uses an onboard flash LED in low-light environments.

Obstacle detection is achieved using dual HC-SR04 ultrasonic sensors with a range of 2–400 cm and millimeter-level accuracy, while fire and smoke hazards are detected using MQ-series gas sensors and infrared flame sensors. Robot movement is controlled through the L298N dual H-bridge motor driver, enabling bidirectional control and PWM-based speed regulation with built-in protection against back-EMF.

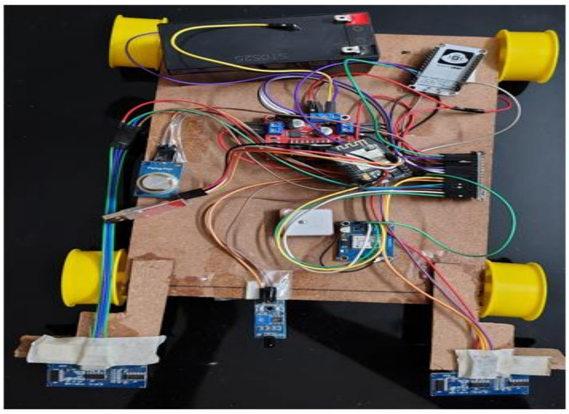

Fig. 3. Complete Hardware Prototype

This work is licensed under a Creative Commons Attribution 4.0 International License

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. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412150

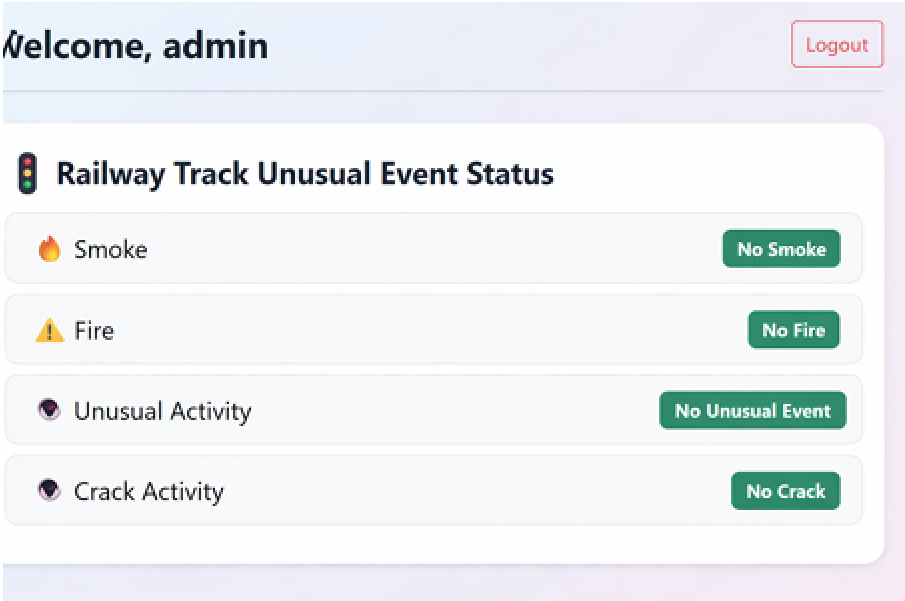

Fig. 4. Web-Based Monitoring Dashboard

This part describes the structure of the software, the development environment, and the implementation of the railway safety monitoring system. The software was created with Python (version 3.8 or higher) and Anaconda which is a great environment for managing packages, isolating environments, and being compatible with multiple platforms such as Windows, Linux, and macOS. The creation of the environment specifically for the machines with vision, data processing, and web development is made easier and more efficient for deployment because Anaconda accompanies that and guarantees reproducibility too. Python was chosen mainly because of its easy-to-understand syntax, vast community support, and rich library ecosystem such as OpenCV which is employed in the development of computer vision and image processing tasks that include but not limited to image preprocessing, edge detection, feature extraction, contour analysis, and Hough transforms. The backend system is built using the Flask web framework which takes care of the communication via RESTful APIs with the IoT devices, routing the requests, exchanging the data in JSON format, and rendering the dynamic web pages.For the purpose of securing user credentials, detection logs, timestamps, system configurations, and historical sensor data, a lightweight SQLite database is utilized, not only securing the data but also making its management and analysis very easy.

VIII. MODULES

8.1 Smart Sensing Module:

This module records tempera ture, humidity, gas concentrations, and particulate matter in real-time. It preprocesses raw signals, detects anomalies, and prepares data for transmission to processing units.

8.2 Vision Processing Module:

ESP32-CAM captures rail way track images, applies edge detection algorithms, and iden tifies potential cracks or defects. Detected anomalies trigger immediate alerts with image evidence.

8.3 Obstacle Detection Module:

Dual ultrasonic sensors provide redundant coverage, measuring distances to objects ahead. The module implements collision avoidance logic and distinguishes between stationary and moving obstacles.

8.4 Wildlife Recognition Module:

Machine learning models process camera feeds to identify animals near tracks. The system classifies detected objects, estimates distance, and generates species-specific alerts.

8.5 Fire Detection Module:

Integrated flame sensors, gas sensors, and computer vision analyze multiple indicators of f ire hazards. Multi-modal fusion reduces false positives while ensuring rapid detection.

DOI: 10.17148/IJARCCE.2025.1412150

8.6 Communication Module:
NodeMCU establishes WiFi connections, formats data as JSON, and transmits to cloud servers via HTTP/MQTT protocols. Implements retry logic and offline buffering for reliability.

8.7 Alert Generation Module:
Cloud-based service evaluates incoming hazard reports, classifies severity, and dispatches notifications via multiple channels (email, SMS, dashboard alerts, mobile push notifications).

8.8 Dashboard Visualization Module:
Web interface dis plays real-time system status, sensor readings, detection events, and historical trends. Administrators monitor fleet health and respond to alerts efficiently.

IX. PERFORMANCE EVALUATION

A. Testing Methodology
System performance was evaluated through controlled lab oratory testing and field deployment trials. Metrics assessed include detection accuracy, response time, false positive rates, and system reliability.

B. Detection Accuracy
1) Crack Detection: Computer vision algorithms achieved:
- True Positive Rate: 94.2%
- False Positive Rate: 3.8%

C. System Response
Time End-to-end latency analysis:
- Sensor Reading: 50-100ms
- Local Processing: 200-500ms
- Wireless Transmission: 100-300ms
- Cloud Processing: 50-150ms
- Alert Generation: 50-100ms
- Total Response Time: 450-1150ms

D. Reliability and Availability
72-hour continuous operation testing:
- System Uptime: 99.2%
- Data Transmission Success: 98.7%
- Sensor Failure Rate: ¡0.5%
- Power Consumption: 4.5W average

E. Cost Analysis Component cost breakdown (approximate):
- Arduino Uno: INR 400 • ESP32-CAM: INR 450 • NodeMCU ESP8266: INR 250
- Ultrasonic Sensors (2×): INR 200
- Gas Sensors (MQ-135, MQ-136): INR 400
- Dust Sensor: INR 600
- Flame Sensor: INR 150
- GPS Module: INR 500
- Power Supply & Misc: INR 800
- Chassis & Motors: INR 1,000
- Total System Cost: INR 4,750

This represents 90-95% cost reduction compared to com mercial automated inspection systems, enabling widespread deployment.

F. Comparative Analysis
Performance comparison with existing approaches:
Results demonstrate that the proposed system achieves superior detection accuracy at significantly reduced cost while maintaining real-time operation capabilities.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2025.1412150

TABLE I
SYSTEM PERFORMANCE COMPARISON

Method	Accuracy	Cost	Real-time
Manual Inspection	75-85%	High	No
Static Cameras	80-85%	Very High	Yes
Drone Systems	85-92%	Very High	Limited
Our System	91-97%	Low	Yes

X. CONCLUSION

The Project demonstrates an IoT technology-based self driven railway safety robot which does not follow the old fashioned ways.The robot identifies the disturbance immediately, for instance, if the rails are broken or there are animals nearby. It also detects signs of fires, blockages and splits as tiny as one millimeter without any delay. The difference between this unit and the manual checklists is that it runs continuously while the latter is walked; thus, the response time is greatly reduced. It is so robust that it can work in a situation where humans allude to. Each time, the risk of accidents is reduced a little bit more.Smoke and fire danger are signaled at an early stage because the air and temperature detection devices are alert all the time. Notifications of got that risks are sent out rapidly via wireless communication, being delivered directly to the screens where the users can see them instantly. The system is modular, thus it can be very easily adapted when new components are added in the future. Moreover, by using very low-cost components, the manufacturing price is kept down even if hundreds of units are produced. Thus, affordability meets practicality and no expensive extras are required.

10.1 Future Work

Several promising directions exist for enhancing system capabilities and expanding deployment scope: 1) Autonomous Navigation Enhancement: Future versions can include GPS-based navigation and camera-based line following to enable independent track movement. Obstacle avoidance and automatic return to the base station during low battery conditions can further improve system reliability. 2) Advanced Detection Capabilities: Detection accuracy can be improved using advanced deep learning models and additional sensors such as thermal cameras and LiDAR. These upgrades will support better identification of track wear, dam aged sleepers, and hazards in low-visibility conditions. 3) Communication Infrastructure: Long-range and high speed communication technologies (5G, LoRaWAN, and mesh networking) can be added to the system. The backup com munication alternatives may also be used to ensure there is connectivity in remote locations. All in all, these improvements are intended to ensure that there is a fully autonomous, scalable, and cost-effective system to inspect railways and enhance safety and efficiency in railways maintenance. The development of the future will focus on such improvements according to the response of field deployments, needs of stakeholders, and technological improvements. The final solution is to establish complete autonomous, dependable, and economical railway inspection facilities that prevent accidents, enhance maintenance effec tiveness, and safety of the transportation system as a whole.

ACKNOWLEDGMENT

The authors would like to acknowledge the Department of Information science and engineering at the, The Oxford College of Engineering who offered laboratory facilities and resources in undertaking this project. We credit also our faculty advisors and technical staff with the help and advice during the formulation of this system.

REFERENCES

[1]. S. Kumar and R. Sharma, "IoT-Based Railway Track Monitoring Sys tem: A Comprehensive Review," *IEEE Access, vol. 9, pp. 45612-45628, 2021.*

[2]. M. Patel et al., "Autonomous Mobile Robots for Railway Infrastructure Inspection," *IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 6, pp. 5124-5138, 2022.*

[3]. A. Singh and V. Kumar, "Multi-Sensor Integration for Enhanced Railway Safety," *Journal of Rail Transport Planning & Management, vol. 18, pp. 203-218, 2021.*

[4]. J. Chen et al., "Ultrasonic Sensor-Based Obstacle Detection for Au tonomous Vehicles: Methods and Applications," *Sensors, vol. 21, no. 8, pp. 2698, 2021.*

[5]. R. Gupta and S. Verma, "Fire Detection Systems in Railway Infrastruc ture: Technologies and Challenges," *Fire Safety Journal, vol. 125, pp. 103-118, 2021.*

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⸎ Peer-reviewed & Refereed journal ⸎ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412150

[6]. L. Wang et al., "NodeMCU ESP8266: A Comprehensive Guide for IoT Applications," *IEEE Internet of Things Journal, vol. 8, no. 15, pp. 11234 11248, 2021.*
[7]. K. Reddy and M. Rao, "Arduino-Based Sensor Networks for Industrial Monitoring," *International Journal of Advanced Research in Computer Science, vol. 12, no. 3, pp. 87-94, 2021.*
[8]. T. Desai et al., "Real-Time Web-Based Monitoring Systems Using Flask and IoT," *Procedia Computer Science, vol. 167, pp. 1234-1243, 2020.*
[9]. P. Brown and J. Smith, "Motor Control Systems for Mobile Robotics: Design and Implementation," *Robotics and Autonomous Systems, vol. 145, pp. 103-119, 2021.*
[10]. V. Sharma et al., "Wireless Sensor Networks for Railway Applications: A Survey," *IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1045-1076, 2021.*
[11]. N. Kumar and A. Jain, "Cost-Effective IoT Solutions for Smart Infras tructure," *IEEE Technology and Society Magazine, vol. 40, no. 3, pp. 67-75, 2021.*
[12]. S. Lee et al., "Performance Analysis of HC-SR04 Ultrasonic Sensor for Distance Measurement Applications," *Measurement Science and Technology, vol. 32, no. 8, pp. 085102, 2021.*
[13]. M. Ahmed and F. Khan, "MQ Series Gas Sensors: Characteristics and Applications in Environmental Monitoring," *Sensors and Actuators B: Chemical, vol. 345, pp. 130-145, 2021.*
[14]. R. Johnson et al., "Real-Time Alert Systems for Critical Infrastructure Monitoring," *IEEE Systems Journal, vol. 15, no. 4, pp. 5432-5443, 2021*