ADVANCED RAILWAY SECURITY SYSTEM FOR TRACK FAULT DETECTION

V Veeresh, Guguloth Vasudev, Vattapa Sam Benny, Burri Sai Kumar, Addepalli Daniel Rajaratnam
Department of Computer Science and Engineering (Internet of Things)
ACE Engineering College, Ankushapur(V), Hyderabad, India-501301

ABSTRACT:

Indian Railways is the largest railway network in Asia and additionally world's second largest network operated underneath a single management. Due to its large size it is difficult to monitor the cracks in tracks manually. This paper deals with this problem and detects cracks in tracks with the help of ultrasonic sensor attached to moving robot. Ultrasonic sensor allows the device to moves back and forth across the track and if there is any fault, it gives information to the cloud server through which railway department is informed on time about cracks and many lives can be saved. This is the application of IoT, due to this it is cost effective system. This effective methodology of continuous observation and assessment of rail tracks might facilitate to stop accidents. This methodology endlessly monitors the rail stress, evaluate the results and provide the rail break alerts such as potential buckling conditions, bending of rails and wheel impact load detection to the concerned authorities. Here we proposed an innovative approach to detect railway track crack as this system detects crack based on image processing. Many image preprocessing steps is used to detect railway track crack. As image is prone to noise. System converts image to grayscale image and uses filtering to remove noise from image. Noise removal helps to detect crack more accurately. Image luminous level is increased and image is converted to binary image. This helps system to detect only crack and helps to remove other unwanted objects. Image once converted to binary image, holes are filled by using image processing method this helps to reject all smaller objects which are not required for crack detection. Crack is detected based on the intensity value. Intensity value is for accuracy.

Keywords: Railway Track Monitoring, Crack Detection, Ultrasonic Sensor, Image Processing, Internet of Things (IoT), Railway Safety System

I.INTRODUCTION

The safety and reliability of railway transportation are of paramount importance, especially in a vast and interconnected network like Indian Railways. The devastating consequences of track failures, such as derailments and accidents, highlight the urgent need for an effective and efficient solution for railway track monitoring. Several key factors have motivated the development of this project. Railway track cracks are among the leading causes of derailments, resulting in significant loss of lives and property. Many of these accidents could be avoided with timely detection and intervention. The motivation to save lives and ensure passenger safety drives the need for an automated crack detection system. Traditional methods of railway track inspection rely on manual processes, which are not only time- consuming but also prone to human error. Given the extensive length of the railway network, these methods are inefficient and inadequate to meet modern safety standards. Automating the process ensures continuous and reliable monitoring, addressing the shortcomings of manual inspections.

The rapid advancements in IoT and image processing technologies offer new opportunities to address long-standing challenges in railway safety. The ability to integrate sensors, cloud connectivity, and advanced algorithms into a single system enables precise and real-time detection of track defects. Harnessing these technologies for railway safety is both practical and forward-thinking.

II. OBJECTIVES

The primary objective of this project is to develop an automated railway track monitoring system that ensures real-time detection of cracks and defects to enhance railway safety. By integrating ultrasonic sensors and advanced image processing techniques, the system aims to achieve high accuracy in identifying track faults while eliminating irrelevant data. Leveraging IoT technology, the solution transmits real-time alerts to a cloud server, enabling timely intervention by railway authorities. The project focuses on creating a cost-effective, scalable, and robust solution that can operate effectively in diverse environmental conditions, contributing to proactive accident prevention and the modernization of railway safety infrastructure.

III.PROBLEM STATEMENT

The Indian Railways, being the largest railway network in Asia and the second largest globally under a single management, faces significant challenges in maintaining the integrity of its extensive tracks. Manual monitoring of track conditions is not only labour-intensive but also prone to human error, posing a serious risk to safety. The primary problem is the efficient detection and reporting of cracks and other faults in railway tracks to prevent accidents. Traditional methods are inadequate for the scale and complexity of the network. To address this, we propose an innovative approach utilizing a combination of ultrasonic sensors and image processing technology, integrated with IoT systems. This solution involves a moving robot equipped with ultrasonic sensors to detect cracks and transmit real-time data to a cloud server. This automated, continuous monitoring system is designed to improve detection accuracy and provide timely alerts to railway authorities, thereby enhancing safety and operational efficiency. The methodology focuses on processing images to filter out noise, convert them to binary format, and use intensity values for precise crack detection. This approach not only ensures cost-effectiveness but also offers a scalable solution for the ongoing maintenance and safety of the railway infrastructure.

IV. PROPOSED SYSYTEM

The proposed system offers an advanced solution to the problem of detecting cracks in railway tracks using a combination of ultrasonic sensors, image processing techniques, and IoT technology. Here is a detailed description:

Ultrasonic Sensors and Moving Robot: The system employs a moving robot equipped with ultrasonic sensors that travel along the railway tracks. Ultrasonic sensors are ideal for this application as they can detect minute changes in the track's surface and identify cracks that mVay not be visible to the naked eye. The sensors continuously scan the tracks, moving back and forth to ensure comprehensive coverage. When the sensors detect a fault, they generate a signal indicating the presence of a crack.

V.HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- ESP-32 Microcontroller
- Lithium-ion Batteries
- Motor Driver, Motors & Wheels
- GPS Module
  IR sensor
- Ultrasonic Sensor

SOFTWARE REQUIREMENTS:

- Operating system – Windows 7, 8, 10, 11, mac os
- Coding Language – Embedded C

- Arduino IDE Tool

VI. SYSTEM IMPLEMENTATION

The implementation of the Railway Track Crack Detection System involves integrating both hardware and software components to create an automated, real-time monitoring solution. At the heart of the system is the ESP32 microcontroller, which coordinates data from IR sensors, ultrasonic sensors, and a GPS module to detect cracks and obstacles along the railway track. The robot moves along the track using motor drivers and DC motors, continuously scanning for anomalies. When a crack or object is detected within a predefined range, the system halts the robot, activates a buzzer, captures the GPS coordinates, and sends an alert via Wi-Fi using the Telegram Bot API. The software is developed using the Arduino IDE and written in Embedded C, enabling efficient sensor data processing, motor control, and wireless communication. Real-time alerts include precise location links to aid maintenance teams in swift response and repair. This implementation showcases a reliable, cost-effective solution that enhances railway safety by reducing dependency on manual inspections and ensuring continuous, accurate fault detection.

VII.CODE USED

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <TinyGPS++.h>
#include <HardwareSerial.h>

// IR Sensors and Buzzer
#define LEFT_IR 34
#define RIGHT_IR 35
#define BUZZER 23

// Motor Pins
#define IN1 14
#define IN2 27
#define IN3 26
#define IN4 25
#define ENB 13
#define ENA 12

// Ultrasonic Sensor
#define TRIG_PIN 32
#define ECHO_PIN 33

// GPS
TinyGPSPlus gps;
HardwareSerial ss(1);  // GPS on Serial1 (D16=RX, D17=TX)
```

```
// WiFi & Telegram
const char* ssid = "Dan";
const char* password = "12345678";
String telegramBotToken = "7871404015:AAEaHx61k6Bsh19OemSnN4hTlFQ5wQxKddE";
String chatId = "1696763494";
bool imageSent = false;
bool objectDetected = false;

void setup() {
  Serial.begin(115200);
  Serial.println("Booting...");

  ss.begin(9600, SERIAL_8N1, 16, 17);  // GPS serial

  pinMode(LEFT_IR, INPUT);
  pinMode(RIGHT_IR, INPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  digitalWrite(BUZZER, LOW);
  stopMotors();

  Serial.print("Connecting to WiFi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("\nWiFi Connected!");
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
```

```
bool left = digitalRead(LEFT_IR);   // 0 = white, 1 = black
bool right = digitalRead(RIGHT_IR);
Serial.print("IR Readings -> Left: ");
Serial.print(left);
Serial.print(" | Right: ");
Serial.println(right);

readGPS(); // update GPS data

float distance = getUltrasonicDistance();
Serial.print("Ultrasonic Distance: ");
Serial.print(distance);
Serial.println(" cm");

if (distance > 0 && distance <= 20 && !objectDetected) {
  objectDetected = true;
  Serial.println("⚠ Object detected within 20 cm. Sending alert.");
  digitalWrite(BUZZER, HIGH);
  delay(1000);
  digitalWrite(BUZZER, LOW);

  float lat = gps.location.lat();
  float lng = gps.location.lng();
  String alertMsg = "⚠ Object detected within 20 cm!\nLocation: https://www.google.com/maps/place/" + String(lat, 6) +
"," + String(lng, 6);
  sendTelegramMessage(alertMsg);
  delay(10000);  // wait before allowing next alert
}

if (distance > 20) {
  objectDetected = false;
}

if (left == 1 && right == 1) {
  Serial.println("Both sensors on line: Moving forward.");
  moveForward();
  imageSent = false;
} else {
  Serial.println("Line break detected: Stopping and alerting.");
  stopMotors();
  if (!imageSent) {
```

```
      alertTheft();
      imageSent = true;
    }
  }

  delay(500); // Short delay to stabilize readings
}

float getUltrasonicDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH, 30000);  // Timeout at 30 ms
  float distance = (duration * 0.0343) / 2;  // in cm
  if (duration == 0) return -1;  // No reading
  return distance;
}

void moveForward() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 200);
  analogWrite(ENB, 200);
  Serial.println("Motors: FORWARD");
}

void stopMotors() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 0);
  analogWrite(ENB, 0);
  Serial.println("Motors: STOPPED");
}
```

```
void readGPS() {
  while (ss.available() > 0) {
    gps.encode(ss.read());
  }

  if (gps.location.isValid()) {
    Serial.print("GPS Location: ");
    Serial.print("Lat = ");
    Serial.print(gps.location.lat(), 6);
    Serial.print(" | Lng = ");
    Serial.println(gps.location.lng(), 6);
  } else {
    Serial.println("GPS location not valid yet...");
  }
}

void alertTheft() {
  Serial.println("🚨 THEFT DETECTED! Triggering buzzer and sending alert.");
  digitalWrite(BUZZER, HIGH);
  delay(2000);
  digitalWrite(BUZZER, LOW);

  float lat = gps.location.lat();
  float lng = gps.location.lng();

  String alertMsg = "🚨 Theft Detected!\nLocation:https://www.google.com/maps/place/" + String(lat, 6) + "," + String(lng, 6);
  sendTelegramMessage(alertMsg);
  sendTelegramPhoto();
  delay(10000);
}

void sendTelegramMessage(String message) {
  HTTPClient http;
  http.begin("https://api.telegram.org/bot" + telegramBotToken + "/sendMessage");
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  String payload = "chat_id=" + chatId + "&text=" + message;
  Serial.println("Sending message: " + message);
  int httpCode = http.POST(payload);
```

```
  Serial.print("Telegram HTTP Response code: ");
  Serial.println(httpCode);
  if (httpCode > 0) {
    String response = http.getString();
    Serial.println("Response: " + response);
  } else {
    Serial.println("Failed to send message");
  }

  http.end();
}

void sendTelegramPhoto() {
  Serial.println("Capturing and sending photo from ESP32-CAM...");

  HTTPClient http;
  String url = "http://ESP32-CAM-IP/capture";  // Replace with actual ESP32-CAM IP
  Serial.println("Image URL: " + url);
  http.begin(url);
  int httpCode = http.GET();
  Serial.print("Photo capture HTTP code: ");
  Serial.println(httpCode);
  http.end();
}
```

VIII.TESTING AND EVALUATION

The testing and evaluation phase of the Railway Track Crack Detection System was crucial to ensure its accuracy, reliability, and real-time responsiveness. The prototype was tested under controlled conditions to simulate real railway track scenarios. During testing, the robot successfully detected artificial cracks and nearby obstacles using its ultrasonic and IR sensors. The system’s response to these detections was evaluated by observing its ability to stop movement, trigger the buzzer, capture GPS coordinates, and send alert messages via Telegram. The GPS module accurately provided location data, which was verified through Google Maps links. The evaluation also included testing different track surfaces, lighting conditions, and object sizes to ensure the image processing and sensor detection were not affected by environmental factors. The robot consistently delivered correct outputs without false alerts, demonstrating the robustness of the system. Overall, the testing confirmed that the system can effectively perform real-time monitoring, accurate fault detection, and timely alert generation, making it suitable for real-world deployment in railway safety operations.

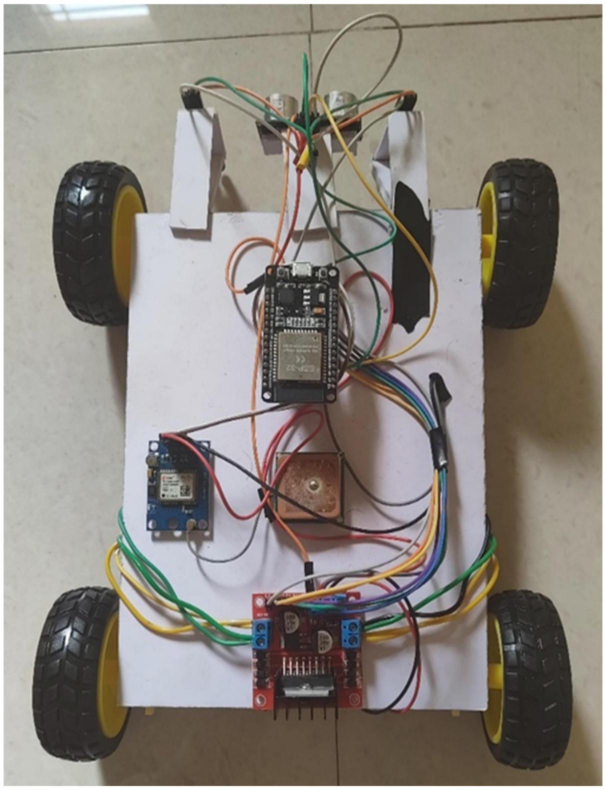

Figure 1: Robot Train

Figure 2 : Showing about Crack Location and Alert

IX. CONCLUSION

The proposed Railway Track Crack Detection System demonstrates a practical and cost-effective solution to one of the most critical challenges in railway transportation—track safety. By integrating ESP-32 microcontroller, ultrasonic and infrared sensors, and a GPS module, the system is capable of efficiently detecting cracks and providing precise location-based alerts in real time. The use of a mobile application for instant notifications enhances the responsiveness of maintenance teams, reducing the chances of accidents due to unnoticed track faults. This project not only showcases the potential of sensor-based protocols. The prototype confirms that real-time monitoring and quick response systems can drastically improve maintenance efficiency and passenger safety. With further development and integration into existing systems, this technology can contribute significantly to building a safer and smarter railway network.

X. REFERENCES

1. Lad, Pranav, and Mansi Pawar. "Evolution of railway track crack detection system." 2016 2nd IEEE international symposium on robotics and manufacturing automation (ROMA). IEEE, 2016.
2. Srivastava, Saurabh, et al. "Railway Track Crack Detection Vehicle." International Advanced Research Journal in Science, Engineering and Technology 4.2 (2017): 145-148.
3. Kumar, Maneesh, et al. "A survey on crack detection technique in railway track." 2018 Conference on Emerging Devices and Smart Systems (ICEDSS). IEEE, 2018.