Railway Track Crack Detection System Using ESP32 + ML on Laptop

# Overview

This project is designed to detect cracks in railway tracks using a smart robotic car that moves along the tracks and uses sensors (IR and Ultrasonic) to collect data. The data is transmitted to a laptop over Wi-Fi, where a machine learning model is run. If a crack is detected, the laptop commands the ESP32 robot to stop, spray paint on the crack, and send updates to a webpage.

# Why This Approach?

## Traditional Method (IR + Ultrasonic only):

- Detects obvious cracks using raw thresholds (e.g., IR value < 100).

- Fails for very slim or hidden cracks.

- Cannot adapt or learn from new environments.

- No prediction or pattern learning.

## ML-Based Method (Sensor + AI):

- Collects real-time sensor data and uses trained ML model.

- Can detect even minor variations and learns patterns of crack behavior.

- Improves over time with more data.

- Reduces false positives and negatives.

# Hardware Setup

Main Components:

- ESP32: Brain of the robot, collects and sends data over Wi-Fi.

- IR Sensors (TCRT5000): Detects surface contrast (dark line/crack vs bright rail).

- Ultrasonic Sensor (HC-SR04): Measures distance from track to detect sudden dips.

- Motor Driver (L298N): Controls DC motors for movement.

- Servo Motor: Used to trigger paint spraying.

- Buzzer + LED: Alert indicators.

- Wi-Fi: ESP32 connects to laptop server.

- Laptop: Runs a Python server and ML model.

# Circuit Connections

|  |  |
| --- | --- |
| Component | ESP32 GPIO Pin |
| IR Sensor 1 OUT | GPIO4 |
| IR Sensor 2 OUT | GPIO16 |
| Ultrasonic Trig | GPIO13 |
| Ultrasonic Echo | GPIO12 (via voltage divider) |
| Motor ENA | GPIO22 |
| Motor IN1 | GPIO21 |
| Motor IN2 | GPIO19 |
| Motor IN3 | GPIO5 |
| Motor IN4 | GPIO18 |
| Servo Signal | GPIO23 |
| Buzzer | GPIO15 |
| LED | GPIO2 |

# ESP32 Code Explanation

- Reads sensor data (IR1, IR2, Ultrasonic).

- Sends data to the laptop in CSV format via HTTP POST.

# Laptop Side Code Explanation

- Python Flask server receives sensor data from ESP32.

- Data is saved to CSV file for training/testing.

- Trained ML model is loaded for real-time predictions.

- If crack is predicted, it sends commands to ESP32 to stop, reverse, spray paint, and alert.

# ML Model Training

Input Data Format:

ir1, ir2, ultrasonic, crack

Example: 1, 1, 7.8, 0

Model: MLPClassifier with structure:

- Input → 16 neurons (first hidden layer)

- Hidden → 8 neurons

- Output → 1 neuron (crack or not)

# UI and Performance

- Simple Flask Web Dashboard: Displays incoming sensor data and crack detection results.

- Logs all data to CSV.

- Fast ML inference on laptop (few milliseconds).

# Why This is Better

Comparison Table:

Feature | Traditional | ML-based  
--------|-------------|---------  
Detect fine cracks | No | Yes  
Adapt to environment | No | Yes  
Improve over time | No | Yes  
Alert + Action | Partial | Full  
Web Reporting | No | Yes

# Future Improvements

- Add camera and train deep learning model.

- Use time series models like LSTMs.

- Integrate GPS for crack location.