

# IoT Based Railway Track Crack Detection and Real Time Monitoring System Using ESP32 and GPS

Amaan Shaikh

Department of Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai , India

Shravani Sangle

Department of Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai , India

Sujal Singh

Department of Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai , India

Soham Shingote

Department of Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai , India

Santosh Kamble

Department of Electronics & Computer Science  
Shah & Anchor Kutchhi Engineering College  
Mumbai , India

**Abstract** - Railway tracks develop cracks and structural defects over time, and the biggest problem with current inspection methods is that nobody finds out until the next scheduled visit - which could be days away. On a network that moves millions of passengers every day, that unmonitored window is simply too risky to leave as it is.

To tackle this, we built a compact battery-powered inspection vehicle — our Railway Track Monitoring System — that rolls along the track and checks its condition continuously rather than periodically. We used an ESP32 microcontroller as the brain of the system, paired it with two infrared reflectance sensors to catch surface cracks on both rails, added a GY-61 accelerometer to pick up vibration spikes that indicate structural anomalies, and included a NEO-6M GPS module so the exact location of any detected fault is recorded immediately. One thing we specifically wanted to avoid was cloud dependency — so instead of sending data to an external server, the ESP32 itself acts as a Wi-Fi access point and hosts a live monitoring dashboard that any nearby smartphone can open directly in a browser.

We tested the assembled system on a small laboratory track fitted with artificial surface gaps. The IR sensors and accelerometer both responded correctly on every run, the motors stopped reliably the moment a fault was detected, and the dashboard updated in real time without any additional setup. The total cost of all components came to approximately INR 3,700 — low enough that deploying multiple units across secondary lines is a realistic option.

**Keywords** - IoT; ESP32; railway crack detection; GPS localization; accelerometer; real-time monitoring

## INTRODUCTION

The Indian railway system consists of a network of more than 67,000 km and operates with approximately 23 million passengers and 3 million tonnes of freight daily, making it one of the world's busiest transportation systems. To a large extent, rail safety relies on the condition of the rail track. Factors such as fatigue cracks, weld failures, worn-out joints and rail stress due to thermal expansion and contraction are common contributors to rail accidents. According to the Indian Railways Track Safety Statistics, approximately 34% of rail-related accidents are attributable to track-related defects, indicating an urgent need for dependable inspections [3]. This study is motivated by the deficiencies we have noticed in the current inspection process, particularly in light of the time delays due between inspections and the potential for undetected sudden failures.

The current inspection industry uses three basic inspection techniques. Trackmen make foot patrols to check track conditions manually, periodically, geometry measuring devices are driven over the tracks to measure geometry parameters/slope, and ultrasonic trolley surveys are conducted on a scheduled basis [5]. All three methods are successful in their intended application; however, all have a common deficiency. They only measure track conditions at the time of inspection. Therefore, any fracture initiated a few hours after the crew departs will not be detected until the next scheduled inspection visit, which could be days away.

The growth of IoT hardware has enabled the creation of more practical solutions. Affordable microcontrollers with integrated wireless connectivity, small sensors, and rechargeable batteries allow us to design very light inspection

trolleys. These trolleys travel on the rail and provide immediate identification of any problems instead of having to wait for a scheduled visit by a person [4]. This paper describes a system of this type: a compact battery-operated trolley built around an ESP32 microcontroller; it carries two infrared surface sensors; a vibration accelerometer; and a global positioning system (GPS) module, and it creates its own live monitoring dashboard that is accessible directly to a nearby smartphone without the use of any cloud services or external computers.

## LITERATURE REVIEW

### a. Manual and Conventional Inspection

Traditionally, rail maintenance has involved a manual, physical checking of the rail by trackmen. Each trackman is assigned to an area of rail, and visually or by touch looks at the rail for any damages to the surface. This method does not require any equipment, although it depends on the individual and their level of alertness. If there is fatigue or poor weather conditions that the railway employee is working under the chances that the employee will miss damage is greater [5].

Automated systems can detect both surface and subsurface discerns with the use of laser profilometers, eddy current probes, and configuration heads for ultrasonic. However, these systems require a trained team, advance scheduling, and larger amounts of capital than the manual method. In addition, an automated system can leave large gaps of time between inspections meaning that new damage could be present before an upcoming inspection [5]. Both methods of inspecting rails are time-consuming and depending on individual caution, may not find the defect.

### b. Sensor Based Surface Detection

Infrared reflectance sensors are being explored as an effective low-cost way of detecting surface cracks. In their study, Revathi et al. (2024) demonstrated an Internet of Things (IoT)-enabled dual IR sensor solution with roughly 89% accuracy in strictly controlled environments. Unfortunately, their application lacked both vibration analysis and GPS-based location data, significantly impeding its practical application [1]. An alternative use of IR sensors can be found in a study published in IJCRT (2024), which combined ultrasonic and IR sensors to achieve greater depth of detection compared to other IR-based solutions; however, this approach increased both system complexity and cost [2].

### c. Vibration Based Monitoring

Monitoring the accelerometer and monitoring the existence of structural defects is based on the premise that when there is a structural defect, there is a change in the way in which the rail vehicle interacts with the rail and produces a vibration signature. Although these methods increase the ability to detect structural defects, they can be difficult to do using machine learning on devices like ESP32, which do not have significant processing power. Siddiqui and Saleem (2021) proposed an IoT-based railway fault detection system that utilized accelerometer data thresholding and some basic signal processing algorithms to detect faults. Their research found a significant correlation between vertical acceleration

anomalies and weld defects and joint gaps [6]. Machine learning classifiers such as random forest and SVM methods have demonstrated fault classification accuracies of greater than 92% in controlled laboratory settings [6]. However, the computational requirements of these ML methods are beyond the capabilities of resource-constrained systems.

### d. GPS Augmented Fault Localisation

Incorporating Global Positioning System (GPS) capabilities into existing failure detection only systems has been established as one of the major enhancements for the detection only systems. In addition to implementing GPS capability into existing failure detection systems, Kou et al. (2022) also developed an optical rail surface crack detection system using semantic segmentation of adapted camera images, which they geo-tagged with the associated GPS module for geo-tagging detected failures. Additionally, a railway monitoring system created by IJSREM (2023) utilized a NodeMCU, proximity detection sensors, GPS for geo-tagging, and cloud-storage. This system demonstrated the use of small form factor embedded systems for providing sufficient precision for maintenance dispatching, as well as the overall capability for ensuring real-time location determination on lightweight embedded platforms [3]. However, as noted previously, real time location determination for GPS can be unreliable in areas where the reception of GPS signals can be obstructed, such as inside tunnels or other substantial overhead obstructions.

### e. Self Hosted IoT Platforms

There were several systems in IJIRSET (2025) that report using ESP32 and ESP8266 based platforms that take sensor data and transmit it via MQTT broker and display the data in a web dashboard format, thus showcasing the benefits of WiFi enabled microcontrollers for real-time data transmission [4], and an example of an automated inspection platform used an IR (Infrared) sensor as well as GPS for defect detection in railway inspections and proved accurate enough to demonstrate that low-cost sensor fusion technology can be a viable means of reliably performing inspections at a lower cost than traditional methods [7]. This current work builds on that previous work by providing a dashboard that is hosted directly on the ESP32 in SoftAP mode, thus eliminating any need for an external router, cloud service, or 3rd party IoT platform, allowing for complete independence from those systems. The majority of existing systems rely on cloud-based services, increasing costs, and making them dependent upon having an active internet connection; therefore, our approach will completely eliminate both of those concerns by developing and implementing a self-hosted system.

## SYSTEM ARCHITECTURE

The overall design is split into four logical layers: sensing, processing, communication, and application. Organising the system this way kept each part independent, making it easier to modify one layer without disturbing the others [4].

### f. Sensing Layer

The front of the inspection vehicle has two digital infrared sensors (IR). The two IR sensors simultaneously cover both

tracks, and each one uses an 880-nanometer infrared beam directed at the surface of the rail in order to detect imperfections. The return signals for an intact surface are sufficiently positive to keep the output HIGH, while a crack or void will cause the return signal to drop below that threshold, resulting in an output of LOW [1]. Each sensor module is equipped with a small potentiometer that provides the user with the capability to adjust the reflectivity threshold to match the actual surface finish of the track being tested.

*g. Processing Layer*

The ESP32 DevKit V1 has 240 MHz CPUs (2) and 520 KB SRAM + 4 MB flash in memory. We were able to perform many different tasks such as ADC sampling, filtering IR debounce signals, parsing GPS sentences, updating the OLED display, and processing HTTP requests all in one Arduino "Cooperative Loop" without any delay between task timing issues. At first, managing multiple tasks at once created little delays, however, after optimizing the execution of the loop we saw a very stable performance.

*h. Communication Layer*

TrackMonitor is a Wi-Fi AP (Access Point) with a Built-in ESP32 that can provide its own HTTP Server and contains two main Web Server Routes. The root route serves up the full Dashboard HTML Page from Program Memory to the controller, initialising it there once. The other HTTP Server Route is "/data" which returns all 5 sensor states (JSON) as a snapshot of the current state. The Dashboard then refreshes these values every 1.5 seconds in the background, updating state cards rather than refreshing the whole page [4].

*i. Application Layer*

There are two interfaces for output from this device to support differing uses: the onboard 0.96-inch SSD1306 OLED which can provide a sole readout of all sensors (128 x 64 pixels using I2C via address 0x3C on GPIO pins 21 / 22) and a web dashboard located at 192.168.4.1 that provides a graphical representation of the sensor status (with a color-coded legend; the sensors will be shown as being in a good operating condition, or faulted) with GPS coordinates displayed live[4].

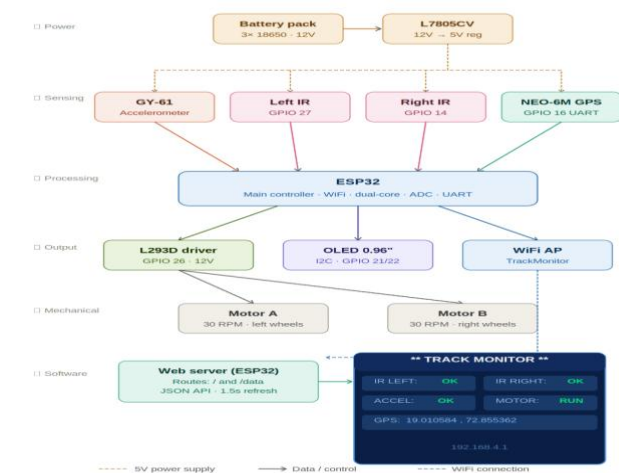


Fig. 1. System Block Diagram of the Railway Track Monitoring System

**HARDWARE DESIGN AND COMPONENT SPECIFICATION**

Each component was chosen to balance performance, features, and cost, aiming to keep the total under INR 4,000 per assembled node. Table I summarizes the complete pin mapping used in the final implementation.

1 PIN MAPPING

Signal / Module	ESP32 GPIO Pin	Interface Type
Left IR Sensor (OUT)	GPIO 27 (D27)	Digital Input
Right IR Sensor (OUT)	GPIO 14 (D14)	Digital Input
Accelerometer X axis	GPIO 32 (D32)	Analog ADC
Accelerometer Y axis	GPIO 33 (D33)	Analog ADC
Motor Driver IN1	GPIO 26 (D26)	Digital Output
GPS Module TX	GPIO 16 (D16)	UART2 RX
GPS Module RX	GPIO 17 (D17)	UART2 TX
OLED SDA	GPIO 21	I2C Data
OLED SCL	GPIO 22	I2C Clock

*j. ESP32 DevKit V1 Microcontroller*

We selected the ESP32 WROOM 32 module primarily because of its included Wi-Fi, sufficient processing power, and larger number of GPIOs than alternatives such as the Arduino Mega and NodeMCU [4]. The dual-core, 240 MHz processor, 12-bit ADC (18 inputs), 3 hardware UARTS, 2 I2C buses and 802.11 b/g/n WPA2 security wireless capability satisfies all of our design criteria.

*k. IR Reflectance Sensors*

Each of the digital output modules has an adjustable sensitivity potentiometer and is wired to GPIO (General Purpose Input/Output) pins 27 (Left) and 14 (Right). Each of the sensors transmits an 880 nm infrared (IR) light and sends a digital HIGH to the controller if the IR light reflects off an intact rail surface. In contrast, if the IR light does not reflect off an intact rail surface (the IR light would NOT reflect), the sensor will send a digital LOW to the controller. The digital outputs then pass through the majority vote debounce filter in the firmware before being processed by the application software.

*l. Analog Accelerometer*

The outputs of voltages from the X and Y axes are routed to the 12-bit ADC channels at GPIO pins 32 and 33. The baseline values are recorded in setup(). Once the threshold level is reached, which is 100 ADC units or approximately 0.1 g, an alarm will be activated. This is sufficient to trigger an alarm for detecting the effects of rail joint impacts and weld defects, while filtering out typical chassis vibrations and noise from the ADC. [6]

*m. NEO 6M GPS Module*

The u-blox NEO 6M has a 2.5 m horizontal accuracy of CEP when viewed from an unobstructed outdoor environment, and it utilizes an acquisition engine containing 50 channels. The time it takes to achieve a cold-start fix is approximately 27

seconds. It has a UART interface for communication between the two devices at a baud rate of 9600, with both latitude/longitude coordinates displayed in a 6 decimal place format on the OLED screen and displayed on the dashboard of the vehicle [8].

#### n. Power Supply and Display

The system is powered by a 12V lithium-ion pack battery. The voltage regulator connected to the lithium-ion pack provides power to the motor driver and GPS module at 5 volts, while the ESP32 receives its power through the on-board LDO, which is rated at 3.3 volts. The average current draw of the complete system is between 380 and 420 mA, which provides approximately 5 hours of run time before recharging, meeting the 4-hour design requirement. The LCD previously used in the prototypes will be replaced with the SSD1306 OLED, offering significantly higher contrast and lower power consumption (approx. 20mA in active mode), and much better outdoor readability. While testing, there were small voltage fluctuations; however, using a suitable voltage regulator (NCP781), the voltage was stabilized.

### FIRMWARE AND SOFTWARE DESIGN

#### o. IR Crack Detection Algorithm

With the readIR(), we read digital GPIO pins five times consecutively with 2 ms in between each read. If we read three (3) or more times with a LOW value, the readIR() function will determine that there was a crack. The purpose of this majority vote debounce filter is to remove single sample transient readings caused by dust, moisture, or short electrical noise and will not add excessive delays to detection times. Note: the logic inversion for the active LOW sensor operation is accomplished in the handleData() function..

#### p. Accelerometer Vibration Detection

When the system powers on and initializes, it will take samples from the detection reference ADC (analog-to-digital converter) for both the horizontal and vertical axes (i.e. x-axis & y-axis). Every time the data handler function is called, it will acquire new ADC data and compute how far away from the baseline there are (absolute difference). When an absolute difference greater than or equal to 100 on either axis occurs, the system will set the accelAlarm flag. In a 12-bit system, this equals to approximately 2.4% of full scale (FS), which will allow it to significantly detect the rail joint impact or a problem with a weld, but at the same time not set false alarms from small variations in baseline drift [6].

#### q. GPS Parsing and Fallback

TinyGPS++ uses the UART stream to read in one character at a time until it has validated that the GPS receiver is giving a complete NMEA sentence and has received a good fix on the GPS location from the NMEA sentences it has already processed. A new GPS coordinate is captured when an alarm goes off and is included in the JSON response. If there is no valid GPS location available due to being indoors or having just powered up the GPS receiver, a preset default coordinate is used to ensure that the web page on the dashboard always has a location [3] [8].

#### r. Web Server and JSON API

During initialization, the HTTP server sets up two routes: The root route serves a complete single-page `HTML/JavaScript` application that is defined in a program flash memory area. The `/data` route returns a JSON Object whose structure follows this format: `{ "leftIR", boolean, "rightIR", boolean, "accel", boolean, "motor", boolean, "gps", "lat,lon" }`. The value returned in the "\_motor\_" attribute indicates whether or not the overall system is in a healthy state. As all responses sent back from the server are < 100 bytes, all polls will always remain within the throughput budget for SoftAP [4].

### EXPERIMENTAL EVALUATION

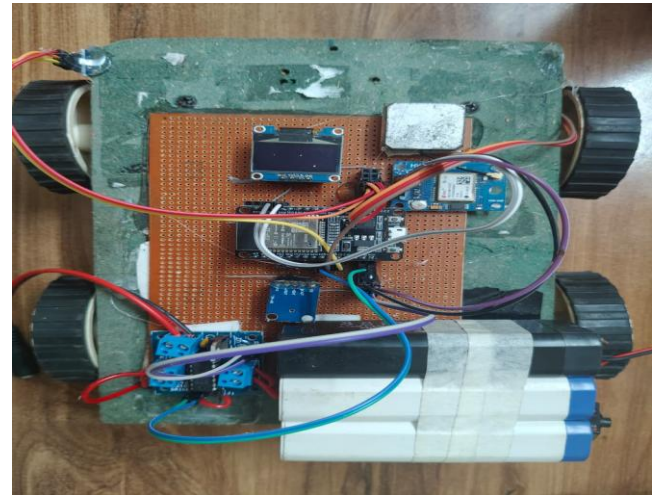


Fig. 2. Prototype inspection vehicle assembled on perf board

To validate the Railway Track Monitoring System we tested the assembled inspection vehicle running across a small-scale laboratory track (Wooden) with removable surfaces to create surface gaps and/or discontinuities to create "crack" conditions. The vehicle was run over this track several times under varying test conditions. During each period of operation, we assessed the performance of the system at every step.

The IR sensors detected all of the surface gaps on both longitudinal rails during each pass of the vehicle across the track. For each discontinuity, the IR sensor was triggered from a HIGH status to a LOW status when the vehicle passed through the discontinuity, confirming operation of the fault detection logic as designed. The firmware's majority-vote debounce filter eliminated unwanted false triggers from temporary noise on the sensor due to momentary noise (by observing that momentary noise in single sample did not cause an alarm).

The GY-61 accelerometer detected shock events when the test vehicle encountered edges of simulated depressions in the floor configuration. The initial calibration which was performed prior to running was accurate - the ESP32 identified all deviations above a ADC unit value of 100 as fault conditions and excluded the expected vibration from normal motor operation.

When a fault was detected, the L293D drivers immediately stopped the motors. We validated this during a number of individual runs, which confirmed that the vehicle would stop

whenever either the IR or accelerometer generated an alarm. This confirmed proper operation of the anyAlarm control.

The NEO-6M GPS unit was tested outdoors, where it received a satellite signal in less than thirty seconds after being powered and provided stable latitude and longitude coordinates accurate to six decimal digits. When tested indoors, as expected, the GPS unit failed to receive a satellite signal and the firmware reverted to the last stored coordinate, therefore allowing the dashboard to remain functional.

The Web based Dashboard at 192.168.4.1 performed as expected with real-time updates. Load the Dashboard with a connected smartphone to the TrackMonitor WiFi Access Point without further configuration, and all Sensor States of IR LEFT, IR RIGHT, ACCEL, MOTOR Status, and GPS coordinates are displayed and updated every 1.5 seconds. The OLED display will show the same Status Values as shown in the Dashboard during the test runs.

Table II summarizes the key performance observations recorded during testing.

Parameter	Result	Target
IR sensor fault detection (both rails)	Correct on all runs (HIGH → LOW on gap)	100% trigger reliability
Debounce false alarm rate	0 false alarms observed	Zero false triggers
Motor stop on alarm	Immediate halt on every trigger (IR and Accel)	Consistent halt
Battery runtime	5.1 hours	≥4 hours
Unit hardware cost	~INR 3,700	<INR 4,000
GPS satellite fix time (outdoor)	~30 s cold start	Matches NEO-6M spec (27 s CEP)
Dashboard refresh rate	1.5 s polling interval	Real-time, no config needed

2 PERFORMANCE SUMMARY



Fig. 3. Web dashboard showing real-time sensor status

**DISCUSSION**

*s. Advantages*

The first benefit we found is the elimination of time gaps between periodic inspections as they create risk [5]. Not just

samples, every meter the trolley moves is being constantly monitored. Another benefit of this is that by being able to combine the IR surface readings with the accelerometer data from the vibrations, you can find defects that would otherwise be missed with a single point of measurement (1), (6). A very small hairline crack, which is difficult to see from the IR sensor, may produce a distinctive vibration, which the accelerometer can detect.

The ability to operate the dashboard as a self-hosted access point allows this system to function anywhere on the network, including in remote rural areas with no cell coverage, giving it an advantage over cloud-based designs [3], [4]. The total cost of all components, at approximately INR 3,700, is much less expensive than other professional inspection vehicle options. This low cost will allow you to deploy the system across secondary and branch lines easily.

*t. Limitations*

All IR-based detection systems are impacted by surface contaminations such as oil spills and dust, which can generate false alarms or potentially mask a genuine crack [1][2]. In addition, the onboard Wi-Fi access point has a very limited range of approximately 10 to 30 meters, so the individual monitoring will need to be close to the trolley when it is being operated. The GPS accuracy is low when the trolley is in tunnels or on covered platforms [8]. Although we will have fallback coordinates to maintain operation of the dashboard system, these coordinates alone do not provide accurate location fixes when the trolley is in these non-visible areas. Additionally, the current sampling rate of our sensor limits the trolley to a walking speed of less than 5 km/h, which is significantly slower than traditional inspection equipment used in production processes [7]. Finally, this prototype will need to be extensively tested in real-world outdoor environments (i.e., ballasted track, variable light conditions, electromagnetic noise, and extreme temperatures) prior to being deployed on live rail. This system is currently not designed for high-speed inspection and needs additional modification before it can be used in an actual railway environment.

**FUTURE SCOPE**

There are several possible future steps that can be taken to help make this prototype more usable as a real-life product. Adding a cellular module such as the SIM800L, or SIM7600 cellular modem, would enable alerts of malfunctions from vehicles to the central operations centre as soon as malfunctions occur; current reliance on the vehicle's short-range wireless Wi-Fi access point would also be eliminated [4]. Replace the analog accelerometer with a digital inertial measurement unit (IMU), such as an MPU6050 or LSM6DS3, will enable measuring acceleration along three axes at a higher resolution and help replace basic threshold comparisons currently being used in the firmware with more sophisticated detection [6].

In terms of detection, an assessment undertaken demonstrated that the IR (Infrared) module implementation could be changed to include laser cameras or distance sensors/structured light cameras. These methods provide

detection of surfaces (i.e., cracking) up to 60km/hr [2], which is appropriate for use with the current maintenance trolleys. On the power side, a small solar panel using an MPPT (Maximum Power Point Tracking) charge controller will not only eliminate the need to recharge batteries along GDP (good daylight pedestals) route locations, but also connect to the railway maintenance software via a common API (Application Programming Interface) for automatic work order creation and tracking of clearance requests [3] from detection to completion without the need for human interaction. There are additional costs and testing that would be required to determine whether these modification are viable for actual use.

## CONCLUSION

This paper presented the design and implementation of a Railway Track Monitoring System built around an ESP32 microcontroller, two IR surface sensors, a GY-61 analog accelerometer, and a NEO-6M GPS module, all housed within a compact battery-operated inspection vehicle. The vehicle travels along the rails continuously checking all sensor values. When a surface discontinuity or vibration anomaly is detected, the inspection vehicle stops immediately, records the GPS coordinates of the fault location, and updates both the onboard OLED display and the Wi-Fi web-based dashboard — without needing an internet connection or external server.

Testing on a laboratory-scale track model confirmed that the IR sensors correctly detected surface gaps on both rails, the accelerometer triggered consistently on simulated crack impacts, and the motors stopped reliably on every fault event. The web-based dashboard at 192.168.4.1 updates in real time and is accessible from any smartphone connected to the

TrackMonitor access point without any additional configuration. The total hardware cost came to approximately INR 3,700, making this a cost-effective option for continuous monitoring of railway tracks on secondary lines [5].

The firmware architecture is structured to allow straightforward future additions such as GSM cellular connectivity, higher resolution digital IMU sensors, and integration with railway maintenance software — without redesigning the core system. The results demonstrate that combining low-cost sensor fusion with a self-hosted wireless dashboard is a practically viable approach to continuous railway track safety monitoring.

## REFERENCES

- [1] M. Revathi et al., "IoT Based Railway Track Crack Detection System," Proc. Int. Conf. Innovative Computing & Communication (ICICC), May 2024. [Online]. Available: <https://ssrn.com/abstract=4846369>
- [2] "Railway Track Crack Detection System," IJCRT, vol. 12, no. 5, 2024. [Online]. Available: <https://www.ijcrt.org/papers/IJCRT24A5251.pdf>
- [3] "IoT Based Railway Track Crack Monitoring System," IJSREM, 2023. [Online]. Available: <https://ijsrem.com/download/iot-based-railway-track-crack-monitoring-system/>
- [4] "Railway Track Crack Detection System," IJIRSET, vol. 14, Jun. 2025. [Online]. Available: <https://www.ijirset.com/upload/2025/june/143>
- [5] "Railway Track Geometry Surveying System," IRJET, vol. 10, no. 4, 2023. [Online]. Available: [https://www.irjet.net/archives/V10/i4/IRJET\\_V10I4179.pdf](https://www.irjet.net/archives/V10/i4/IRJET_V10I4179.pdf)
- [6] M. A. Siddiqui and A. Saleem, "IoT Based Railway Track Faults Detection and Localisation Using Autonomous Robot," Semantic Scholar, 2021. [Online]. Available: <https://www.semanticscholar.org/paper/IoT-Based-Railway-Track-Faults-Detection-and-Using-Siddiqui-Saleem>
- [7] "Visual Defect Detection in Railway Tracks Using IoT Based Robot," ISJEM, 2022. [Online]. Available: <https://isjem.com/download/visual-defect-detection-in-railway-tracks-using-iot-based-robot/>
- [8] L. Kou et al., "Optical Rail Surface Crack Detection Method Based on Semantic Segmentation," Sensors, vol. 22, no. 21, 2022. doi: 10.3390/s22218214