AI-Powered Railway Track Inspection Robot

Prof. Revathi B, Prof. Reshma Farhin J

Leela Selvamathi B, Monisha Y M, Saranya R, Sri Madhura S
Department of Artificial Intelligence and Machine Learning
Sri Sairam College of Engineering, Bengaluru, Karnataka

Abstract*—Track-level defects such as cracks, joint gaps, misalignment, sleeper damage, and foreign-object intrusion pose a direct risk to safe train movement. Conventional inspection practice still relies heavily on trained personnel walking the line or on dedicated inspection vehicles, both of which are slow to deploy, costly to operate, and difficult to scale across long route lengths. This work presents a compact, low-cost robotic prototype, the AI-Powered Railway Track Inspection Robot, that combines embedded motor control, machine vision, and inertial sensing on a single mobile platform. An ESP32 microcontroller drives a 3D-printed miniature train over a model track using geared BO motors and a dual-channel motor driver, while an onboard MPU6050 six-axis sensor continuously reports acceleration and rotational rate so that vibration, shock, and tilt events can be flagged as the train advances. A forward-facing USB camera streams footage to a companion laptop, where an OpenCV pre-processing stage feeds a YOLO-based detector trained to recognize cracks, rail gaps, obstacles, missing rail sections, sleeper faults, and alignment errors. A Wi-Fi link lets the laptop issue an immediate stop command to the ESP32 the moment a defect is confirmed by the AI pipeline, and this software-level safety path is backed up by a dedicated hardware emergency button and buzzer that operate independently of the wireless connection. A browser-based dashboard renders the live camera feed together with sensor readings, AI detection results, and running totals of inspection rounds and logged faults. Because the stop/continue decision draws on both the camera output and the inertial readings rather than either signal alone, the platform is less likely to miss a fault that a single sensing channel would overlook on its own. The resulting prototype offers a workable, low-cost template for AI-assisted rail-infrastructure monitoring that can be carried from an academic demonstrator toward early-stage field trials.*

Index Terms*—Railway track inspection, machine vision, YOLO object detection, ESP32 microcontroller, MPU6050 inertial sensor, multi-sensor fusion, vibration monitoring, IoT-enabled robotics, emergency-stop system, live monitoring dashboard.*

I. Introduction

Rail infrastructure is under constant mechanical and environmental stress from repeated wheel loading, vibration, and weathering. Left unaddressed, this stress gradually produces surface cracks, gaps at rail joints, misaligned sections, sleeper degradation, and other localized irregularities. Delayed detection of such faults both raises long-term maintenance cost and increases the operational risk to trains running over the affected section. Because conventional inspection is carried out only periodically, either by manual walkover or by specialized survey vehicles, a defect can develop unnoticed for a considerable stretch of time between inspection cycles.

A growing body of inspection research addresses this gap by combining optical sensing, inertial measurement, and learning-based analysis. Cameras are effective at picking out visually apparent defects and obstructions, whereas accelerometers and gyroscopes capture vibration, impact, and tilt behavior that can reveal an irregularity even when it is not obviously visible to a camera. These two sensing families have largely been explored on their own in earlier work: autonomous and robotic platforms for rail-maintenance tasks [1], combined IMU-and-vision approaches to track-geometry measurement [2], camera-based embedded systems for surface-deformation detection [3], and YOLO-family detectors applied to rail-surface and fastener defects [4]–[7].

The AI-Powered Railway Track Inspection Robot (KS5583) described here draws on these separate threads and merges camera-based defect recognition with IMU-based vibration/tilt sensing on a single low-cost, miniature, mobile unit.

Because real-time YOLO inference is far more computationally demanding than motor control or sensor polling, the vision workload is deliberately offloaded to a laptop rather than run on the ESP32 itself. The ESP32 is instead kept dedicated to time-critical tasks — motor actuation, emergency-stop handling, buzzer control, MPU6050 sampling, and Wi-Fi messaging — while the laptop carries out AI inference and serves the monitoring dashboard. This division of responsibilities keeps the architecture simple and mirrors a pattern that is practical for cost-effective inspection robots more generally.

II. Literature Survey

Vithanage, Harrison, and DeSilva [1] surveyed more than seventy publications on robotic and autonomous systems used in railway maintenance and concluded that mobile robotic platforms are being adopted for inspection tasks at a growing rate; the survey, however, stops short of proposing any specific onboard sensing suite or AI pipeline, a gap that the present prototype fills in a low-cost context.

Escalona, Urda, and Muñoz [2] built a track-geometry measurement rig that pairs an inertial measurement unit with video cameras and multibody kinematic modeling; their results indicate that combining inertial and visual data yields more dependable measurements than relying on either channel alone, although their platform is designed as a full-scale track-geometry car rather than a portable, low-cost prototype.

Memon, Memon, Kalwar, and Chowdhry [3] developed an embedded inspection prototype that pairs a vision camera with an accelerometer and time-synchronizes contour-based surface-fault detection with the recorded vibration signature — an approach conceptually close to the fusion strategy adopted here — but their system does not incorporate a trained AI detection model or provide a remote monitoring dashboard.

Luo, Cai, and Li [4] fine-tuned an enhanced YOLOv5s architecture for rail-surface defect detection and reported accuracy strong enough to justify using a YOLO-family model for rapid defect recognition, while Zhang, Xu, Zhang, and Deng [5] paired an image-enhancement stage with an improved YOLOX detector specifically to counter poor illumination, underlining how important robust image pre-processing is for a detector intended for continuous, real-world deployment.

Li, Wang, Yang, Wang, and Zhang [6] adapted a modified YOLOv5s network to locating defective rail fasteners, showing that lightweight single-stage detectors generalize beyond rail-surface cracking to other railway components. Traquinho et al. [7] mined axle-box acceleration data with data-science methods to flag early-stage track damage, reinforcing the case for treating onboard acceleration measurement as a meaningful indicator of track health, and Xia, Han, and Kwon [8] investigated synthetic image generation as a way to offset the shortage of real defect images available for training railway surface-defect detectors.

Read together, these studies confirm that camera-based defect recognition, inertial vibration analysis, and embedded or robotic delivery platforms have each received substantial individual attention, yet very few published systems combine a mobile ESP32-controlled chassis, laptop-hosted YOLO inference, MPU6050-based vibration/tilt sensing, an independent hardware emergency stop, and a live web dashboard into one affordable, demonstrable unit. Closing this combined gap is the specific contribution of the AI-Powered Railway Track Inspection Robot presented in this paper.

III. Problem Identification

- Manual inspection of rail track is slow and repetitive, and its accuracy depends on how consistently an individual inspector can spot defects, particularly across long stretches of line.
- A vision-only system can give unreliable results under poor lighting, motion blur, or track geometry that departs from the expected pattern.
- A vibration-only system may flag that something is wrong without indicating what the underlying defect actually looks like, which complicates fault verification.
- Much of the equipment used for track inspection today is specialized and costly, making it hard to scale to frequent, low-cost monitoring.
- There remains a need for a compact, portable platform that can traverse a track, sense faults through more than one modality, halt automatically when danger is detected, and relay the fault to a live monitoring interface.

IV. Limitations of Existing Systems

A. Reliance on Manual Observation

Inspection practice today is still largely built around periodic walkovers or scheduled vehicle surveys, which means a developing fault can go unnoticed for the entire interval between two inspection visits.

B. Single-Modality Sensing

A system built around vision alone struggles under poor lighting or motion blur, while a system built around vibration alone cannot show what is causing the anomaly it detects, leaving less room for confident fault verification.

C. High Cost and Limited Scalability

As with specialized sensing equipment in other industries, purpose-built rail-inspection hardware tends to be costly and difficult to roll out at the scale that frequent monitoring of an extensive network would require.

D. Lack of Integrated Safety Response

Detection accuracy tends to dominate the research literature, while comparatively little attention goes to building an integrated, independent mechanism that can bring a moving inspection platform to a stop the instant a dangerous condition, or a manual emergency signal, is registered.

V. Proposed System

- 3D-Printed Train and Track Platform — a reusable miniature mechanical chassis that carries all onboard electronics and lets defect scenarios be repeated on demand.
- ESP32-Based Motion Control — actuates BO geared motors through a dual H-bridge driver, giving forward, reverse, stop, and variable-speed control.
- Camera -Based Defect Recognition — a front-mounted USB webcam feeds live video to a laptop running OpenCV and a YOLO-based model that identifies cracks, rail gaps, obstacles, sleeper damage, and misalignment.

- MPU6050 Inertial Sensing — derives vibration and tilt indicators from measured acceleration and angular rate, complementing what the camera observes.
- Sensor-Fusion Decision Logic — weighs camera confidence together with IMU indicators, with the emergency input always taking precedence, to reach a stop/continue decision.
- Emergency Stop and Buzzer Alert — a physical button and buzzer deliver hardware-level safety response and audible alerting independent of software.
- Live Web Dashboard — presents the inspection video feed, AI detection confidence, sensor readings, completed-round counts, and the cumulative fault tally.

Functionally, the system is organized into three cooperating layers: a physical layer made up of the 3D-printed train, track, motors, camera, battery, emergency button, and buzzer; an embedded-control layer built around the ESP32, motor driver, and MPU6050; and an AI/monitoring layer consisting of the laptop that runs OpenCV/YOLO inference and hosts the dashboard. The overall interconnection among these layers is summarized in Fig. 1: the laptop and ESP32 exchange status and command messages over Wi-Fi so that an AI-confirmed critical defect can trigger an immediate motor-stop instruction, while the emergency button and the core safety logic remain available locally on the ESP32 regardless of the state of the wireless link.

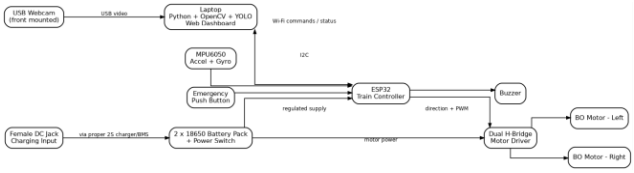

Fig. 1. *System-level block diagram showing the interconnection of the USB webcam, laptop (Python/OpenCV/YOLO/dashboard), MPU6050, emergency push button, ESP32 train controller, dual H-bridge motor driver, BO drive motors, buzzer, and the two-cell battery/charging path.*

VI. System Modules and Methodology

A. Vision-Based Defect Detection

Frames captured continuously by the front-mounted USB webcam are streamed to the laptop, where an OpenCV pre-processing stage prepares each frame before a YOLO-based model classifies it against six defect categories — crack, rail gap, obstacle, damaged or missing sleeper, missing rail section, and misalignment. Detected regions are annotated with bounding boxes, class labels, and confidence scores, all of which are also forwarded to the live dashboard for display.

B. Vibration and Tilt Sensing

The MPU6050 module is fixed rigidly close to the wheel/chassis area so that vibration transmitted through the track reaches the sensor with minimal damping. The ESP32 samples three axes of acceleration and three axes of angular rate, from which it computes the resultant acceleration magnitude $A = \sqrt{(ax^2 + ay^2 + az^2)}$; the deviation $|A - g|$, where g is the gravitational constant, serves as the vibration indicator, while pitch and roll derived from the accelerometer readings are used to flag excessive tilt.

C. Motor and Movement Control

BO geared motors are actuated by the ESP32 through a dual H-bridge driver, giving forward, reverse, stop, and PWM-based speed regulation. Motion commands originate either from manual dashboard input or from the autonomous inspection routine, and either source can be overridden instantly by a stop instruction issued by the AI decision logic, the MPU6050-based safety check, or the emergency button.

D. Emergency Stop and Alert System

The emergency push button is wired as an active-low input using the ESP32's internal pull-up resistor. A single press cuts power to both motors immediately and latches the system into an emergency state until it is manually cleared, and this response does not depend on the laptop connection or on the Wi-Fi link being active. A buzzer sounds whenever a confirmed fault, abnormal vibration, excessive tilt, or an active emergency-stop condition is present.

E. Sensor-Fusion Decision Logic

Rather than averaging the camera and IMU signals together, the stop/continue decision follows a fault-priority scheme in which the emergency input always overrides every other input. An obstacle or rail gap reported by the camera triggers an immediate stop; a visible crack accompanied by abnormal vibration is treated as a confirmed high-confidence fault; a severe IMU vibration reading with no matching camera detection causes the train to stop or slow down and prompts the laptop to save the current frame for manual review; and pitch or roll beyond the configured limit triggers a stop to guard against derailment. Fig. 2 summarizes this priority-based decision flow, showing how the vision result, the IMU result, and the emergency-button state are combined to reach one of three outcomes — normal continuation, a cautionary slow-down/verification response, or a critical stop with buzzer activation and fault logging.

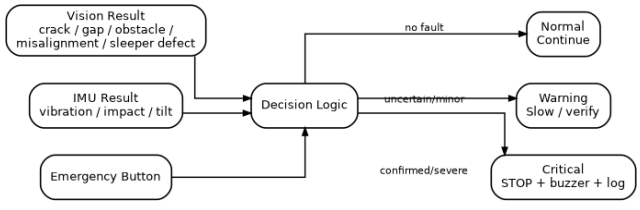

Fig. 2. *Sensor-fusion decision-logic diagram illustrating how vision-based detections, IMU-derived vibration/tilt results, and the emergency-button state are combined into a fault-priority decision that yields a normal, warning, or critical outcome.*

F. Wireless Communication

The ESP32 and the laptop exchange data over Wi-Fi: sensor readings and movement status travel from the ESP32 to the laptop application, while AI-confirmed stop commands

travel back from the laptop to the ESP32, keeping the two subsystems synchronized throughout an inspection run.

G. Web Dashboard Application

A live web dashboard hosted on the laptop shows the camera feed with AI detections overlaid, current stream and inference frame rates, movement and round status, per-detection confidence values, the most recently logged fault with its timestamp, and running totals for completed inspection rounds and cumulative detected faults.

H. Power and Safety System

Onboard power comes from two 18650 lithium-ion cells held in a two-cell holder and routed through a power switch. A female DC jack is provided for external charging, but it must be connected only to a charger/BMS rated for 2S lithium-ion packs — the jack itself carries no protection circuitry and should never be wired directly to the unprotected cells.

VII. Implementation

The design described above was built as a physical miniature train unit mounted on a fixed base plate that carries the complete electronics stack, coupled to a modular 3D-printed circular track. Fig. 4 documents the detailed hardware interconnection used in the build — the specific GPIO pin assignments on the ESP32 for the motor driver, the MPU6050 (I2C), the emergency button, and the buzzer, together with the power path from the two-cell battery pack through the main switch to the regulated supply. This pin-level mapping is the concrete realization of the block-level connections introduced earlier in Fig. 1.

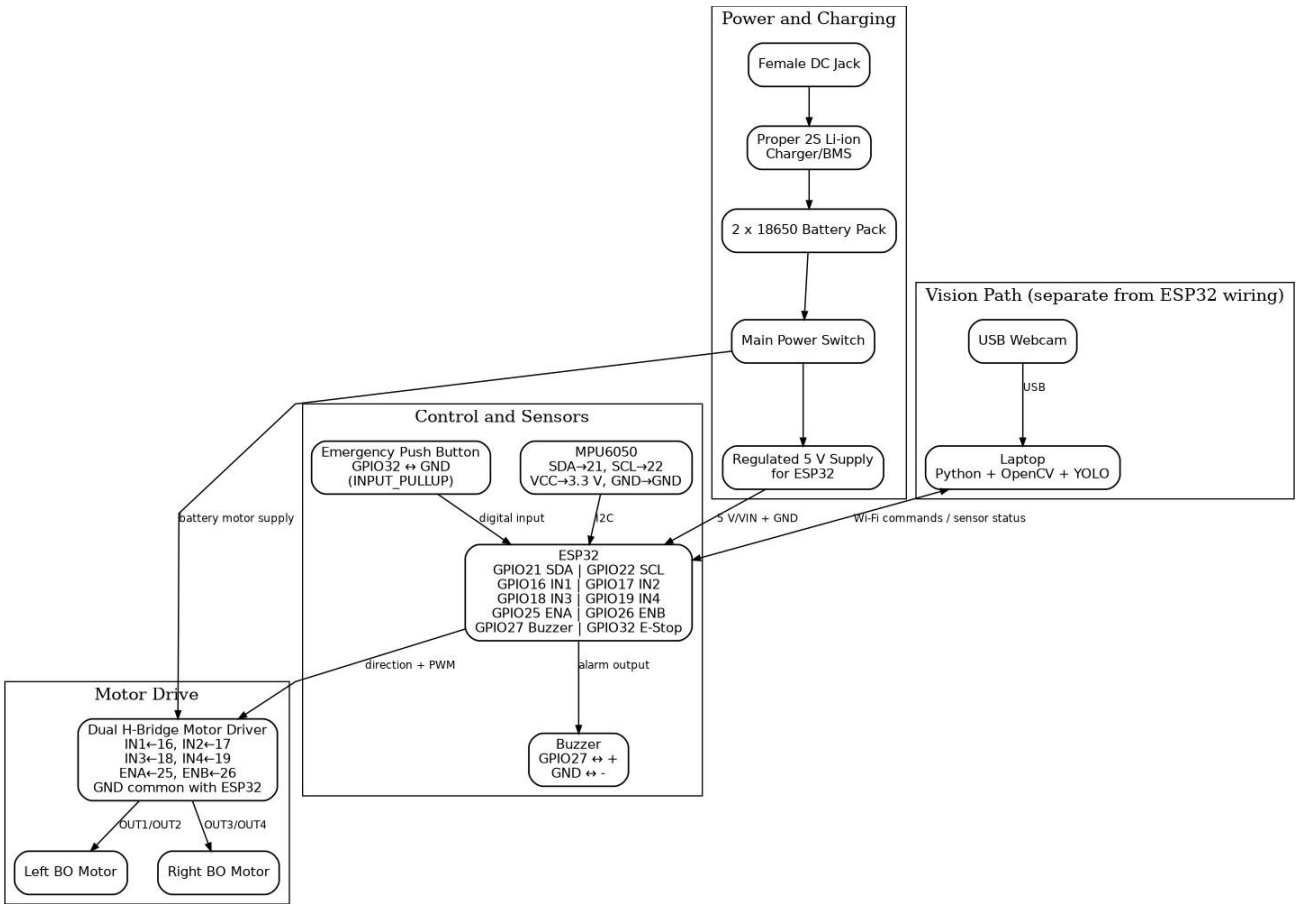

Fig. 4. *Detailed hardware interconnection diagram showing ESP32 GPIO pin assignments for the MPU6050 (I2C), emergency push button, buzzer, and dual H-bridge motor driver, together with the power and charging path from the 18650 battery pack.*

Fig. 3 shows the assembled hardware prototype prior to enclosure, with the ESP32 development board, motor-driver module, buzzer, emergency push button, MPU6050, and the front-facing ball-mounted camera module wired together on the train platform. The wiring harness routes power and signal lines from the battery and driver stage to the controller and sensor modules, consistent with the pin mapping given in Fig. 4.

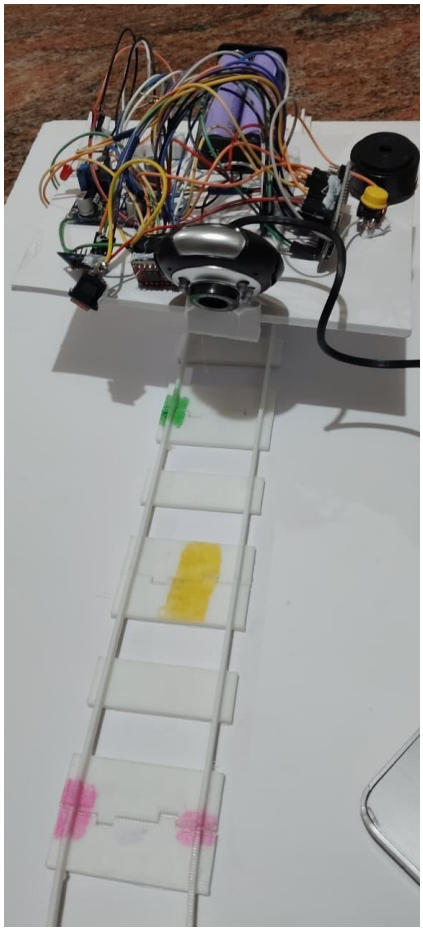

Fig. 3. *Assembled hardware prototype showing the ESP32 controller, motor driver, buzzer, emergency button, MPU6050, and front camera module mounted above the modular 3D-printed track with manually created defect sections (crack, gap, and obstacle markers shown in green, yellow, and pink).*

The modular track sections visible in Fig. 3 include manually created defect zones — a small crack section (green marker), a surface-damage/obstacle section (yellow marker), and a paired rail-gap section (pink markers) — which served as controlled, repeatable test cases for validating the vision pipeline described in Section VI-A. Mounting the camera directly above the leading edge of the track keeps both rails and the sleeper region within the field of view during forward motion, a placement that testing showed to be important for consistent bounding-box localization.

Initial bench testing confirmed that the ESP32 could poll the motor driver, buzzer, emergency button, and MPU6050 concurrently while the laptop independently processed the webcam feed, verifying that the physical wiring and module placement shown in Fig. 3 and Fig. 4 support the complete sensing and control pipeline described in Sections V and VI.

Fig. 5 traces the resulting runtime behavior as a flowchart, from power-on and ESP32/MPU6050 initialization through the continuous loop of frame capture, sensor reading, sensor-fusion decision-making, and either a stop-and-log response or continued inspection, ending in a manual-clear step before the loop restarts.

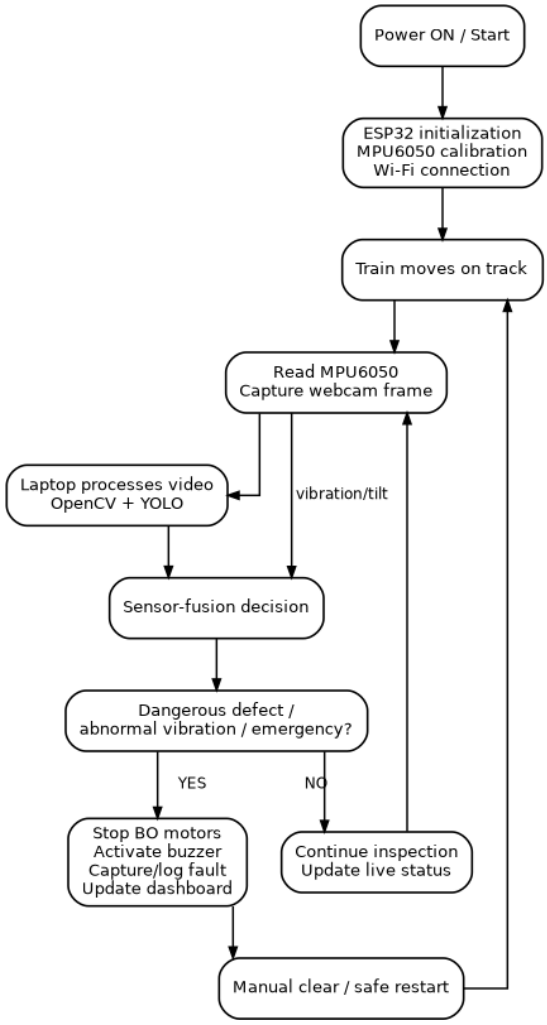

Fig. 5. *Operational flowchart of the inspection cycle, from power-on and sensor initialization through frame capture, sensor-fusion decision-making, fault logging, and manual-clear/restart.*

VIII. Results and Discussion

The end-to-end pipeline was further checked at the application layer using the live inspection dashboard shown in Fig. 6. The dashboard confirms that the camera feed, AI inference pipeline, and inspection-tracking backend function together as one integrated system: the interface reports a live video stream at 7.5 FPS alongside an AI inference rate of about 14.9 FPS, with the loaded detection model (best.pt) shown as “Ready.” Over the recorded session, the system completed 12 inspection rounds and logged 850 total detected faults, each with timestamped evidence.

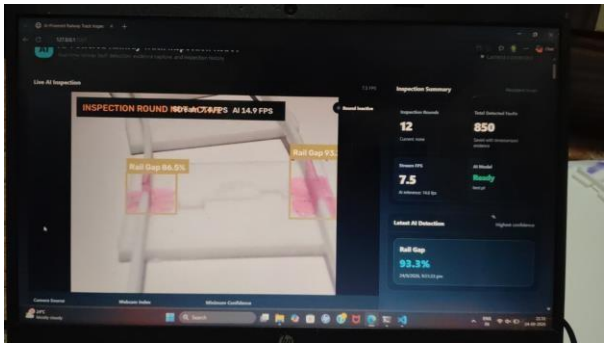

Fig. 6. *Live AI inspection dashboard showing dual rail-gap detections (86.5% and 93.3% confidence), real-time stream/inference frame rates, and persistent inspection totals (12 rounds, 850 detected faults).*

As shown in Fig. 6, the vision pipeline correctly localizes two independent rail-gap defects on the model track, with bounding boxes and confidence scores of 86.5% and 93.3% respectively, and the "Latest AI Detection" panel confirms the higher-confidence rail-gap event along with its timestamp. This confirms that the YOLO-based detector can identify and localize multiple simultaneous defects within a single frame from the prototype's mounted-camera angle, and that detection results are correctly relayed to the dashboard's summary statistics in real time.

A. Detection Reliability and Dataset Dependence

The high confidence values recorded for the rail-gap class in Fig. 6 show that the vision pipeline described in Section VI-A performs well once a defect category is adequately represented in the training data. Whether this same level of reliability carries over to the remaining classes — cracks, obstacles, and sleeper damage — depends mainly on how varied the training dataset is rather than on the detection architecture itself, so future data-collection efforts should prioritize the marked defect zones on the track shown in Fig. 3 under a range of lighting and camera-angle conditions.

B. Vibration-Threshold Calibration

Unlike the vision pipeline, the vibration and tilt indicators computed in Section VI-B cannot be assigned a fixed threshold without empirical tuning, since motor operation, wheel geometry, and the 3D-printed track joints visible in Fig. 3 each add their own baseline vibration. Repeated runs across the marked normal and defect sections are therefore needed to set the severe-stop threshold used by the decision logic summarized in Fig. 2 (Section VI-E), with enough margin to avoid false alarms.

C. Safety Validation

Bench testing confirmed that the emergency push button and buzzer shown in Fig. 3 stop the train independently of the laptop connection, verifying the local-priority safety behavior designed in Section VI-D. Keeping this independence from the Wi-Fi link intact remains the most critical safety property as the sensor-fusion logic in Fig. 2 is tuned further using real vibration data.

IX. Conclusion

The AI-Powered Railway Track Inspection Robot brings together mechanical prototyping, embedded control, computer vision, and inertial sensing on a single miniature railway-inspection platform. A 3D-printed train moves along a model track on BO motors controlled by an ESP32, a front-mounted USB webcam supplies imagery to a laptop for OpenCV- and YOLO-based inspection, and an MPU6050 tracks vibration, impact, and tilt throughout the run. The live dashboard validated in this work shows that the system can sustain continuous inspection rounds, detect track defects such as rail gaps with high confidence, and keep a persistent record of fault statistics for later review. A local emergency push button and buzzer supply essential safety and alerting functions that remain available independent of the wireless link. Taken together, the prototype offers a practical, low-cost route toward AI-assisted railway-infrastructure monitoring, with future work aimed at expanding the training dataset, quantifying detection and vibration-threshold performance more rigorously, and strengthening mechanical rigidity and power-system protection ahead of extended field trials.

Acknowledgment

The authors gratefully acknowledge Prof. Revathi B for her guidance and continued support throughout this project, and thank the Department of Artificial Intelligence and Machine Learning, Sri Sairam College of Engineering, for the resources and facilities that made this work possible.

References

[1] R. K. W. Vithanage, C. S. Harrison, and A. K. M. DeSilva, "Importance and Applications of Robotic and Autonomous Systems (RAS) in Railway Maintenance Sector: A Review," Computers, vol. 8, no. 3, art. 56, 2019.

[2] J. L. Escalona, P. Urda, and S. Muñoz, "A Track Geometry Measuring System Based on Multibody Kinematics, Inertial Sensors and Computer Vision," Sensors, vol. 21, no. 3, art. 683, 2021.

[3] T. R. Memon, T. D. Memon, I. H. Kalwar, and B. S. Chowdhry, "Embedded System Development for Detection of Railway Track Surface Deformation Using Contour Feature Algorithm," Computers, Materials & Continua, vol. 75, no. 2, pp. 2461–2477, 2023.

[4] H. Luo, L. Cai, and C. Li, "Rail Surface Defect Detection Based on An Improved YOLOv5s," Applied Sciences, vol. 13, no. 12, art. 7330, 2023.

[5] C. Zhang, D. Xu, L. Zhang, and W. Deng, "Rail Surface Defect Detection Based on Image Enhancement and Improved YOLOX," Electronics, vol. 12, no. 12, art. 2672, 2023.

[6] X. Li, Q. Wang, X. Yang, K. Wang, and H. Zhang, "Track Fastener Defect Detection Model Based on Improved YOLOv5s," Sensors, vol. 23, no. 14, art. 6457, 2023.

[7] N. Traquinho, C. Vale, D. Ribeiro, A. Meixedo, P. Montenegro, A. Mosleh, and R. Calçada, "Damage Identification for Railway Tracks Using Onboard Monitoring Systems in In-Service Vehicles and Data Science," Machines, vol. 11, no. 10, art. 981, 2023.

[8] Y. Xia, S. W. Han, and H. J. Kwon, "Image Generation and Recognition for Railway Surface Defect Detection," Sensors, vol. 23, no. 10, art. 4793, 2023.

[9] D.-Z. Dang, C.-C. Lai, Y.-Q. Ni, Q. Zhao, B. Su, and Q.-F. Zhou, "Image Classification-Based Defect Detection of Railway Tracks Using Fiber Bragg Grating Ultrasonic Sensors," Applied Sciences, vol. 13, no. 1, art. 384, 2023.

[10] Spherenex Innovation Labs, "AI-Powered Railway Track Inspection Robot (KS5583)," project specification, Quotation Ref. KS5583/SPN-24072026-KS5583, 2026.