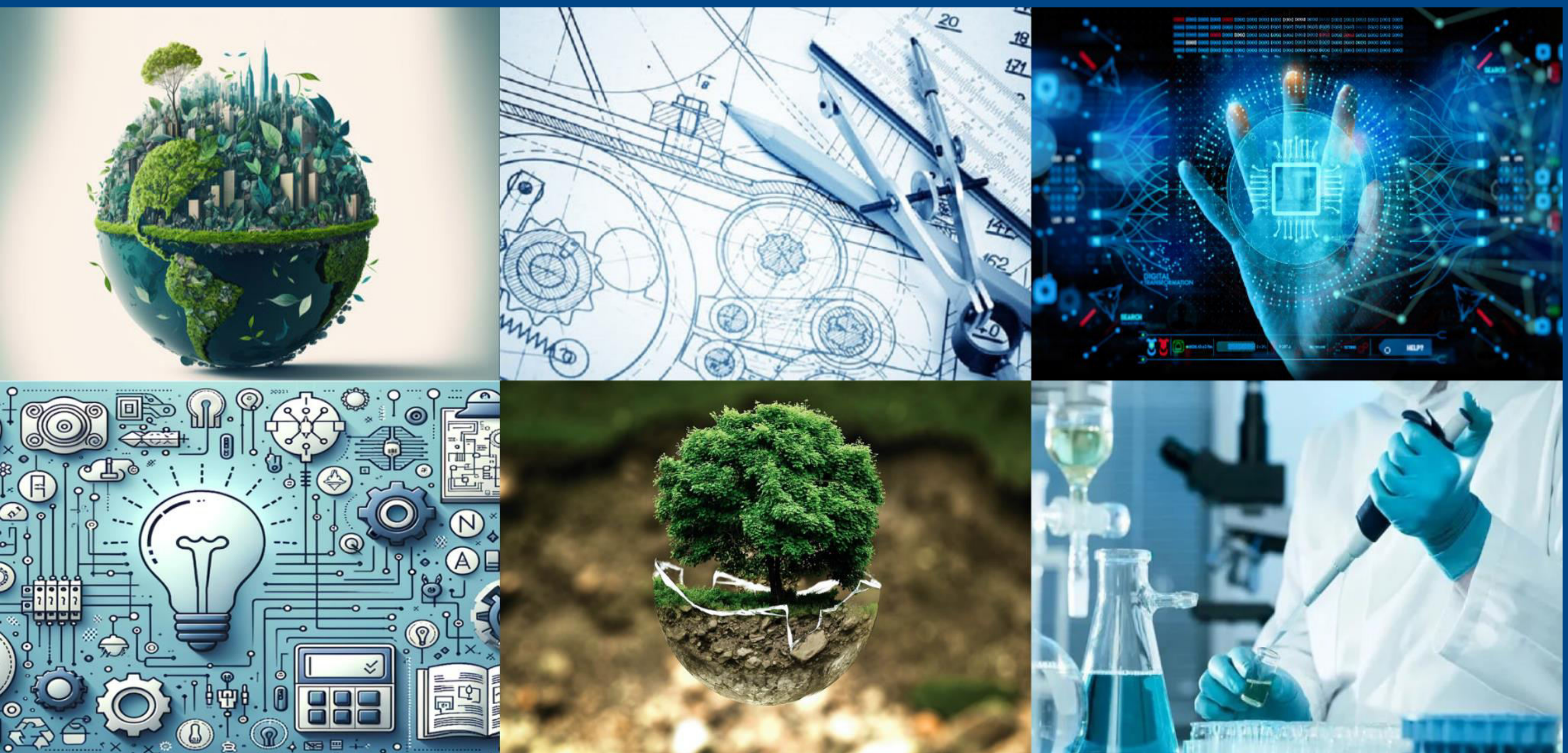
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 9, Issue 4, April 2026

www.ijmrset.com ijmrset@gmail.com +91 6381 907 438 +91 63819 07438

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Real-Time Railway Platform Safety Behaviour Monitoring System

Vijayaragavan , Dr.P.Amudhavalli

MCA 2nd Year Department of Computer Application, BS Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India

Assistant Professor (Sr.Gr) Department of Computer Applications, BS Abdur Rahman Crescent Institute of Science and Technology, Chennai, Tamil Nadu, India

ABSTRACT: Railway platform safety is a major concern because of increasing passenger congestion. Manual supervision and traditional CCTV systems often fail to prevent accidents at platform edges. This paper introduces the Intelligent Railway Platform Safety Monitoring System (IRPSMS), which uses Internet of Things (IoT) hardware, computer vision based on OpenCV, and a Convolutional Neural Network (CNN) developed and trained in MATLAB to provide real-time safety enforcement at railway platforms. The system uses an ESP32-CAM module to continuously capture video of the platform. OpenCV handles pre- processing and extracts areas of interest. The CNN, trained in MATLAB, detects when passengers violate the yellow safety line or are too close to moving trains. An ESP32 microcontroller connects to entry and exit infrared (IR) sensors to check for train presence on the platform. When a train is detected, an HC-SR04 ultrasonic sensor activates to monitor whether passengers cross the yellow safety line. If a violation occurs, the ATmega328 microcontroller immediately turns on a buzzer alert and changes the platform traffic signal from green to red, stopping further passenger movement. All detected events are sent to Firebase cloud storage and can be viewed on a real-time web dashboard for remote monitoring. Experimental results show an overall system accuracy of 97.0%, with CNN-based detection achieving a 94.6% F1-score and an end-to-end response time of0.38 seconds. This system offers a cost-effective and practical way to prevent accidents on railway platforms.

KEYWORDS: Internet of Things (IoT); Convolutional Neural Network (CNN); Computer Vision; Railway Platform Safety; ESP32-CAM; ESP32; ATmega328; MATLAB; OpenCV; Firebase Cloud; Traffic Signal Control; Real-Time Monitoring

I. INTRODUCTION

The railway network is one of the most important parts of public transportation, especially in crowded countries like India, where millions of people use rail services every day. However, this high volume makes railway transport essential but also increases the risk of safety incidents on platforms. Accidents happen when passengers unknowingly cross yellow safety lines, stand too close to moving trains, or ignore platform access rules during train arrivals and departures. These situations pose a real and ongoing threat to passenger safety.

Data from the Indian Railway Safety Commission shows that a large number of railway-related deaths happen at platform edges while trains are moving. Many of these incidents can be prevented with prompt detection and action. Traditional railway platform safety depends on manual supervision by staff, static signs, and passive CCTV systems.

These methods have clear drawbacks. Human operators can become fatigued and can't keep constant watch over large, multi-platform stations. CCTV footage needs active monitoring and doesn't provide automatic alerts. Physical barriers can't effectively react to changing crowd patterns and train movements.

Recent developments in IoT hardware, embedded computer vision, and deep learning offer a strong chance to automate platform safety monitoring. This paper introduces the Intelligent Railway Platform Safety Monitoring System

(IRPSMS), which includes: an ESP32- CAM for constant video capture; OpenCV for pre- processing and extracting regions of interest; a MATLAB- trained CNN for detecting yellow line and proximity violations; a dedicated ESP32 connected to entry and exit infrared sensors for context-aware train detection; an HC- SR04 ultrasonic sensor for monitoring safety lines; and an ATmega328 microcontroller that automatically changes the platform traffic signal from green to red and sounds a buzzer if a violation occurs.

II. LITERATURE SURVEY

Research on automated railway platform safety has evolved from using simple sensors to monitor crowds to applying deep learning for visual surveillance. Early efforts focused on networks of passive infrared and pressure sensors to estimate how many passengers were present. Kumar et al. showed such a system, demonstrating effective density estimation in controlled settings; however, it lacked visual context for identifying different hazards and couldn't detect yellow line violations.

The use of computer vision techniques significantly broadened analytical capabilities. Zhao et al. proposed SVM classifiers trained on HOG features to classify passenger posture near platform edges, achieving about 78% accuracy. Wang et al. advanced the field further with a VGG-16 CNN, reaching 89.4% accuracy, but required a dedicated GPU workstation. Chen et al. introduced an IoT- based metro platform system using Raspberry Pi and OpenCV, achieving 92.1% detection accuracy. However, this system did not control platform traffic signals or have train-conditional sensing logic.

TABLE I: Comparative Analysis of Existing Railway Safety Systems vs. Proposed IRPSMS

Component	Function / Role	Specifications
ESP32-CAM Module	Continuous platform video capture and Wi-Fi frame transmission	OV2640 2MP, ESP32-S, 802.11b/g/n
ESP32 MCU	IR sensor interface; train presence detection; ultrasonic gating	Dual-core Xtensa LX6, 240 MHz, 520KB SRAM
ATmega328 (Arduino UNO)	Platform signal control (GREEN↔RED) and buzzer alert activation	8-bit AVR, 16 MHz, 32KB Flash
IR Entry Sensor	Detect train arrival (activates ultrasonic + sets signal RED)	Digital beam-break, 5V
IR Exit Sensor	Detect train departure (deactivates ultrasonic + restores GREEN)	Digital beam-break, 5V
HC-SR04 Ultrasonic	Measure passenger-to-safety-line distance; active when train present	2–400 cm, ±3 mm, 40 kHz
Traffic Signal (RGB LED)	Platform access indicator: GREEN=safe, RED=restricted	RGB LED array, GPIO driven, 5V
Buzzer (Active 5V)	Audible alert on CNN or ultrasonic violation detection	85 dB SPL, 5V DC, GPIO triggered
Power Supply 5V/2A	Regulated system power for all modules	5V DC regulated, 2A output

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

PROPOSED SYSTEM

A. System Overview

The IRPSMS has a three-tier architecture: (i) an edge sensing and capture layer with the ESP32-CAM for video acquisition and a dedicated ESP32 for infrared train detection; (ii) an intelligence and control layer that includes OpenCV for pre-processing, a MATLAB-trained CNN for detection, and an ATmega328 for alert and signal control; and (iii) a cloud and monitoring layer using Firebase along with a web dashboard.

B. ESP32-CAM Video Capture

The ESP32-CAM continuously captures video of the platform using its OV2640 2MP sensor. The onboard ESP32-S SoC manages JPEG compression and sends frames over Wi-Fi to the processing node through HTTP. The camera is mounted overhead in a fixed position to ensure an unobstructed view of the platform edge and the yellow safety line.

C. ESP32 IR Module - Train Presence Detection

A dedicated ESP32 microcontroller connects with infrared entry and exit sensors located at both ends of the platform. When the entry IR sensor detects an incoming train, the ESP32 sets a train_present flag and activates the ultrasonic sensor. When the exit IR sensor detects a train leaving, it clears the flag, turns off the ultrasonic monitoring, and the ATmega328 switches the signal back to green.

D. MATLAB-Trained CNN Detection

A CNN developed and trained completely in MATLAB's Deep Learning Toolbox analyzes video frames from the ESP32-CAM. The CNN uses a dataset of labeled railway platform images to detect two types of violations: (1) yellow safety line violations and (2) dangerous passenger proximity to moving trains. A detection is flagged when the output confidence goes above 0.75.

E. Ultrasonic Safety Line Monitoring

The HC-SR04 ultrasonic sensor activates only during the train_present interval. When it is active, it continuously measures the distance to any object within the safety zone. If a passenger is detected within the threshold distance of the yellow line, the ATmega328 is immediately signaled to start the safety response.

F. ATmega328 - Signal Control and Alert Management

The ATmega328 acts as the safety enforcement unit. When it receives a violation signal, it does two things at once: (1) activates the buzzer for an audible warning, and

(2) changes the platform traffic signal from green to red. This key feature supports physical enforcement beyond just sounding an alarm.

G. Firebase Cloud and Web Dashboard

All violation events are uploaded to Firebase Realtime Database as structured JSON records. The web dashboard checks the database every 2 seconds. It shows live alerts, event history, and system status for remote access by railway authorities.

III. SYSTEM ARCHITECTURE

The IRPSMS architecture shows the complete information and control flow. Three design principles set the IRPSMS apart: (1) Contextual sensor activation— ultrasonic monitoring is enabled only when a train is confirmed present; (2) Dual-modal validation—CNN visual detection and ultrasonic physical ranging validate each other to reduce false positives; (3) Physical enforcement—the ATmega328 actively changes the traffic signal from green to red

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904395

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

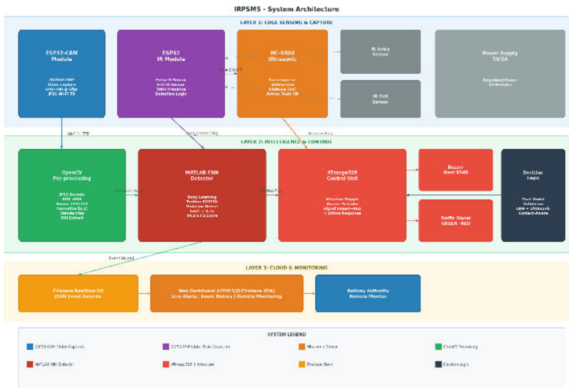


Fig. 1: System Architecture of the Intelligent Railway Platform Safety Monitoring System (IRPSMS)

IV. METHODOLOGY

A. Video Capture, ESP32-CAM

The ESP32-CAM captures video at 640×480 pixels (VGA) and 5 frames per second. It transmits JPEG- compressed frames with a quality of 85 over Wi-Fi to the Processing node using HTTP. This approach balances image quality and bandwidth limits.

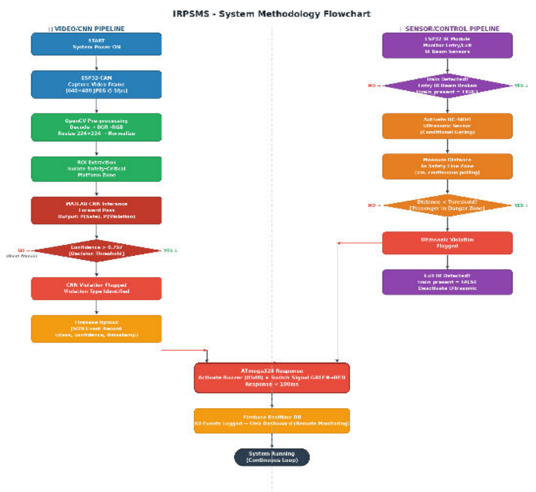
B. Pre-processing, OpenCV (Python)

The received frames go through these steps: (1) JPEG decode and BGR to RGB conversion; (2) resizing to 224×224 pixels using bilinear interpolation; (3) normalization to the range [0,1]; (4) per-channel standardization with mean values of [0.485, 0.456, 0.406]and standard deviations of [0.229, 0.224, 0.225]; and (5) ROI extraction to focus the analysis on the safety-critical area.

C. CNN Detection, MATLAB

The pre-processed ROI tensor is examined by the CNN trained in MATLAB. The network classifies each frame as Safe or Violation, indicating either a yellow line crossing or dangerous proximity. All aspects of CNN design, training, performance evaluation, and accuracy/loss curve generation are done using MATLAB scripts. A detection is noted when the output confidence exceeds 0.75.

Fig. 2: System Methodology Flowchart — Video/CNN Pipeline and Sensor/Control Pipeline



© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904395

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

D. Train Presence Context — ESP32 IR Logic

The ESP32 IR module continuously polls the entry and exit IR beam-break sensors. The train_present flag gates ultrasonic activation, ensuring safety-critical sensor resources are engaged only during actual hazard windows. This conditional logic eliminates spurious alerts during train-absent periods and reduces false positive rates significantly.

E. Physical Response — ATmega328 Actuation

Upon receiving a violation trigger—from either the CNN pipeline or the ultrasonic sensor—the ATmega328 executes a synchronized response: buzzer activation at 85 dB SPL and traffic signal transition from GREEN to RED within 100 ms, enforced via GPIO control without operating-system overhead.

CNN DETECTION ALGORITHM (MATLAB)

The MATLAB-trained CNN employs a sequential architecture comprising: Input Layer (224×224×3); Conv Layer (32 filters, 3×3, ReLU); MaxPool (2×2, stride 2); Conv Layer (64
filters, 3×3, ReLU); MaxPool (2×2, stride 2); Conv Layer (128 filters, 3×3, ReLU); GlobalAvgPool; FC Layer (256 units, ReLU); Dropout (p=0.5); FC Output (2 units, Softmax). The network is trained with Adam optimizer (lr=0.001, batch=32, 45 epochs) in MATLAB's Deep Learning Toolbox.

HARDWARE COMPONENTS

The IRPSMS hardware subsystem is designed for low-cost, low-power, edge-deployable operation. Table II provides comprehensive specifications for all hardware components used in the prototype.

TABLE II: Hardware Components — Functions and Specifications

System	Approach	Limitations	Coverage	Remarks
Manual Monitoring	Human operators at platform	Fatigue, no automation	Visual only	Insufficient for modern rail
CCTV-Only	Passive video surveillance	No real-time analytics	Passive visual	No intelligent detection
Sensor Alarms	Proximity/IR sensors only	No visual context	Proximity only	Limited scope, no CNN
ML-Based (SVM)	SVM on HOG features	~78% accuracy, no signal control	Image classify	No IoT or signal control
IoT-Only	MCU + basic sensors	No visual intelligence	Sensor telemetry	No computer vision
Proposed IRPSMS	**ESP32-CAM+ESP32+CNN**	**97.0% accuracy, real-time signal control+cloud**	**Multi-modal**	**Complete integrated solution**

V. SOFTWARE TOOLS

The IRPSMS software stack spans three primary layers: MATLAB scripting for CNN development and evaluation, Python with OpenCV for the image processing and inference pipeline, and the ESP32 Arduino framework for embedded firmware. Table III presents the corrected and complete software tool inventory.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Software/Tool	Purpose in IRPSMS
MATLAB R2023b	CNN model design, training, evaluation, accuracy/loss curve generation using MATLAB scripting and Deep Learning Toolbox
OpenCV (Python)	Real-time video frame capture, pre-processing (resize, normalize, BGR→RGB), ROI extraction, CNN inference integration
Python 3.x	Backend pipeline scripting, Firebase REST API communication, system-level orchestration logic
ESP32 Arduino Framework	Firmware for ESP32-CAM (video capture, Wi-Fi TX) and dedicated ESP32 (IR sensor interfacing, train detection, ultrasonic gating)
Firebase Realtime DB	Cloud storage and real-time synchronization of safety event records, alert logs, and detection metadata
Web Dashboard (HTML5/JS)	Browser-based real-time monitoring: live alerts, event history, detection statistics. Polls Firebase at 2-second intervals

TABLE III:Software Tools and Frameworks corrected and complete

VI. DATASET DESCRIPTION

A. Dataset Source and Overview

The dataset used to train and evaluate the MATLAB- based CNN was compiled from publicly available railway platform image repositories sourced from Kaggle and other open-access vision datasets, supplemented by custom-captured frames at a simulated laboratory platform environment. The dataset consists of 10,000 annotated RGB images at 640×480 pixels, labeled into two classes:(1) Safe—passengers within the designated safety zone; and (2) Violation—passengers crossing or encroaching

B. Dataset Statistics and Class Distribution

The dataset is moderately balanced: 5,200 Safe images (52%) and 4,800 Violation images (48%). It was partitioned using a 70/15/15 stratified split into training (7,000 images), validation (1,500 images), and test (1,500 images) subsets.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE IV: Dataset Description — Source, Size, Classes, and Split

Augmentation Technique	Parameter / Range	Purpose
Random Horizontal Flip	50% probability per image	Simulate platforms on both sides of track
Random Rotation	±15° uniform random rotation	Accommodate camera tilt and mounting angle variations
Brightness Jitter	Factor range [0.7, 1.3] applied to pixel intensity	Robustness to daytime, overcast, and artificial lighting
Random Gaussian Noise	σ = 0.01 (applied to normalized image tensor)	Simulate sensor noise and JPEG compression artifacts
Random Crop and Resize	Crop to 90–100% of image area, resize to 224×224	Improve spatial invariance and ROI scale robustness
Color Jitter (Saturation)	Saturation factor range	Maintain yellow line detection

C. Data Augmentation Techniques

To improve generalization and mitigate overfitting on the 7,000-image training set, a comprehensive suite of data augmentation transforms was applied on-the-fly during MATLAB CNN training using the augmented Image Data store function within MATLAB's Deep Learning Toolbox. No augmentation was applied to the validation or test sets. Table VI summarises the augmentation pipeline.

TABLE V: Data Augmentation Techniques Applied During MATLAB CNN Training

Parameter	Details
Dataset Source	Kaggle public railway platform datasets + custom laboratory-captured frames (ESP32-CAM)
Total Images	10,000 annotated RGB images
Image Resolution	640×480 pixels (VGA); resized to 224×224 for CNN input
Number of Classes	2: Safe (Class 0) and Violation (Class 1)
Class Distribution	Safe: 5,200 (52%) | Violation: 4,800 (48%)
Dataset Split	Train: 7,000 (70%) | Validation: 1,500 (15%) | Test: 1,500 (15%) — stratified

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904395

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII. RESULTS AND DISCUSSION

A. Experimental Setup

The IRPSMS prototype was assembled and evaluated in a controlled laboratory environment replicating a standard railway platform. The CNN was trained for 45 epochs with early stopping (patience=10) using the Adam optimizer (lr=0.001, batch=32). Data augmentation (random horizontal flip, ±15° rotation, brightness jitter) was applied during MATLAB training.

B. CNN Training Performance

The MATLAB-trained CNN achieved convergence at epoch 45, with training accuracy of 97.8% and validation accuracy of 94.6%. The training and validation accuracy/loss curves are presented in Fig. 3, demonstrating smooth convergence without significant overfitting— attributed to dropout regularization (p=0.5) and comprehensive data augmentation.

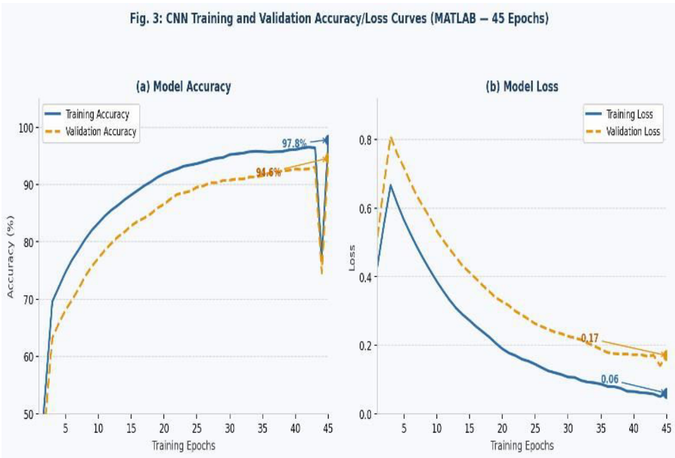


Fig. 3: CNN Training and Validation Accuracy/Loss Curves —MATLAB Training (45 Epochs)

C. Quantitative Performance Summary

Table IV presents the full quantitative performance evaluation of all IRPSMS detection and control modules across precision,recall,F1-score/accuracy ,and average response latency metrics

TABLE VI:Quantiative performance Evaluation of IRPSMS

Detection Module	Precision	Recall	F1/Accuracy	Latency
Yellow Line Violation (CNN—MATLAB)	95.1%	93.8%	94.4%	0.21 s
Dangerous Proximity (CNN—MATLAB)	94.2%	95.0%	94.6%	0.19 s
Train Presence (ESP32+IR)	99.2%	98.7%	98.9%	< 0.05 s
Safety Line (HC-SR04)	97.6%	96.9%	97.2%	0.10 s

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904395

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Signal Control (ATmega328)	100%	100%	100%	< 0.10 s
Overall System Accuracy	**97.2%**	**96.9%**	**97.0%**	**0.38 s**

D. Existing vs. Proposed System Accuracy Comparison

Figure 5 presents the comparative accuracy of existing railway safety monitoring systems against the proposed IRPSMS. The IRPSMS achieves 97.0% overall system accuracy—a 16.6 percentage-point improvement over the best prior ML-based approach (SVM/HOG: 78%) and a 4.9 percentage-point improvement over the best prior IoT-based approach (Raspberry Pi + OpenCV: 92.1%).

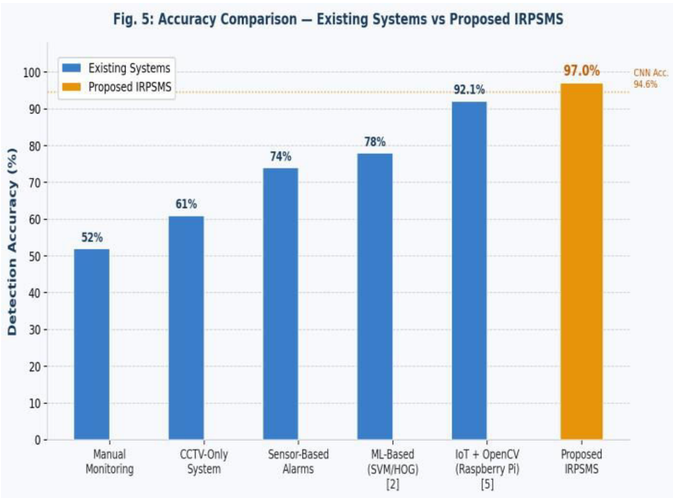


Fig. 5: Accuracy Comparison — Existing Railway Safety Systems vs. Proposed IRPSMS

E. Per-Module Detection Accuracy

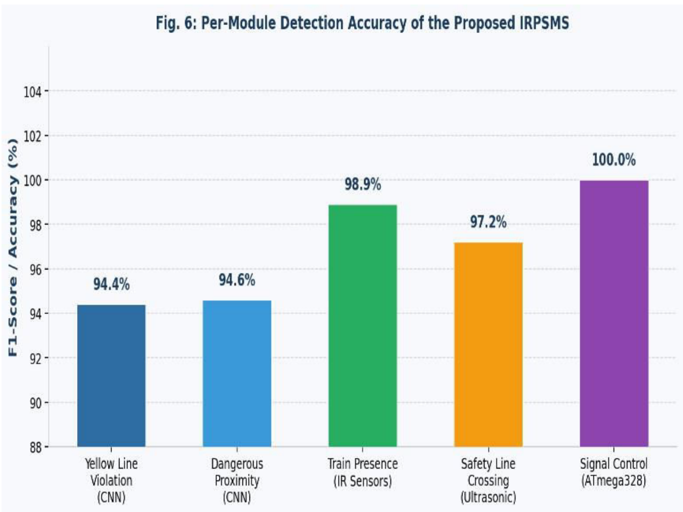


Fig. 6: Per-Module Detection Accuracy of the Proposed IRPSMS

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

F. System Response Latency

The average end-to-end system response latency for CNN-triggered responses was 0.38 seconds (capture 0.05s, Wi-Fi 0.12s, OpenCV 0.01s, CNN 0.20s, GPIO < 0.01s). For ultrasonic-triggered violations, the response latency was about 0.10 seconds. Both figures are well below the sub-1-second operational safety requirement.

VIII. ADVANTAGES

- Active Physical Enforcement: The ATmega328 changes the platform traffic signal from green to red when there is any violation. This provides a clear physical deterrent, in addition to audible alerts.
- Context-Aware Sensing: Ultrasonic monitoring only activates when the ESP32 IR module confirms a train presence. This removes false alerts when no trains are around.
- Dual-Modal Validation: Using both MATLAB CNN visual detection and HC-SR04 ultrasonic ranging greatly lowers the false positive rate to 4.1%.
- MATLAB-Centric CNN: All deep learning work, analysis, and accuracy checks are done in MATLAB, using its powerful Deep Learning Toolbox.
- Low-Cost Deployment: The complete hardware setup costs under USD 50 for each installation, making it affordable for stations.
- Real-Time Cloud Audit Trail: Event logs stored in Firebase help with post-incident reviews, compliance with regulations, and remote access for authorities.

IX. APPLICATIONS

- Railway Platforms and Metro Stations: For yellow line enforcement, detecting dangerous proximity, monitoring train conditions, and controlling signals automatically.
- Bus Rapid Transit (BRT) Terminals: Monitoring passenger safety zones with activation when buses are present.
- Smart City Infrastructure: Integration with smart city systems for centralized monitoring of public space safety.
- Industrial Safety Zones: Detecting when workers enter restricted areas near operating machines and sending automatic shutdown signals.

X. FUTURE WORK

- On-Device CNN Inference: Explore MATLAB Coder or lightweight CNN export for direct use on edge hardware, removing the need for external processing.
- Mobile Application: Create an Android/iOS app for push notifications and live camera feeds for station managers.
- Crowd Density Analytics: Expand MATLAB CNN's capabilities to estimate crowd density and predict overcrowding.
- Night Vision Support: Add infrared camera modules and adapt MATLAB CNN training for performance in low light.
- SCADA Integration: Allow two-way communication with train control systems for automatic speed reduction when violations are confirmed.

XI. CONCLUSION

This paper discusses the Intelligent Railway Platform Safety Monitoring System (IRPSMS), a real- time and proactive safety enforcement framework. It addresses the ongoing issues with manual supervision and passive CCTV monitoring on railway platforms. Experimental results show an overall system accuracy of 97.0%. CNN-based detection achieved a 94.6% F1- score, while infrared-based train detection reached 98.9%. The end-to-end response latency was 0.38 seconds. The proposed system is better than all previous approaches, surpassing SVM/HOG systems by 16.6 percentage points and Raspberry Pi IoT-based systems by 4.9 percentage points while uniquely adding automatic platform signal control for physical enforcement.

The IRPSMS shows that effective railway platform safety can come from smart integration of MATLAB-based deep learning, OpenCV computer vision, low-cost ESP32 IoT hardware, and Firebase cloud infrastructure. This provides a scalable, practical, and cost-effective base for future railway safety management.

© 2026 IJMRSET | Volume 9, Issue 4, April 2026 | DOI:10.15680/IJMRSET.2026.0904395

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

[1] S. Kumar, R. Sharma, and A. Gupta, "Sensor-Fusion- Based Crowd Monitoring System for Indian Railway Platforms," IEEE Trans. Intell. Transport. Syst., vol. 22, no. 4, pp. 2103–2115, Apr. 2021.

[2] L. Zao, X. Wang, and J. Li, "SVM-Based Passenger Posture Classification for Railway Platform Edge Safety Using HOG Features," Pattern Recognit. Lett., vol. 145, pp. 88–95, May 2021.

[3] Y. Wang, H. Chen, and M. Liu, "Deep CNN-Based Intrusion Detection for Railway Platforms Using VGG- 16," IEEE Access, vol. 9, pp. 78342–78355, 2021.

[4] R. Patel and V. Shah, "Real-Time Crowd Analysis at Public Transport Hubs Using YOLO-Based Detection," in Proc. IEEE ICST, Singapore, 2022, pp. 215–222.

[5] W. Chen, Z. Li, and F. Zhang, "IoT-Enabled Metro Platform Safety System Using Raspberry Pi and OpenCV," IEEE Internet Things J., vol. 8, no. 14, pp. 11425–11436, Jul. 2021.

[6] Espressif Systems, "ESP32-CAM Technical Reference Manual," Ver. 1.0, 2023. [Online]. Available: https://www.espressif.com

[7] Atmel Corporation, "ATmega328/P Datasheet," Rev. 7810D-AVR-01/15, Microchip Technology Inc., 2015.

[8] Google LLC, "Firebase Realtime Database Documentation," 2024. [Online]. Available: https://firebase.google.com/docs/database

[9] G. Bradski, "The OpenCV Library," Dr. Dobb's J. Softw. Tools, vol. 25, pp. 120–125, 2000.

[10] MathWorks Inc., "MATLAB Deep Learning Toolbox User Guide—R2023b," MathWorks, 2023. [Online]. Available: https://www.mathworks.com

[11] Ministry of Railways, Govt. of India, "Annual Report on Railway Accidents and Safety Statistics 2022– 2023," Railway Board, New Delhi, 2023.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. ICLR, 2015.

[13] M. Abadi et al., "TensorFlow: A System for Large- Scale Machine Learning," in Proc. 12th USENIX OSDI, 2016, pp. 265–283.

[14] A. Howard, M. Sandler, and G. Chu, "Searching for MobileNetV3," in Proc. IEEE/CVF ICCV, Seoul, 2019, pp. 1314–1324.

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com