

IoT Based Remote Surveillance for Animal Tracking near Railway Tracks using Raspberry Pi

Maddikayala Sai Swetha¹, N. Sukumar²

1. M.Tech Student, Dept. of ECE Jayamukhilnstitute of Technological Sciences, Warangal, India
maddikayalawetha9@gmail.com
2. Assistant Professor, Dept. of ECE Jayamukhi Institute of Technological Sciences, Warangal, India
Sukku.0477@gmail.com

ABSTRACT

Railway accidents involving animals pose a significant threat to both wildlife and train operations, leading to fatalities and disruptions. This project proposes an IoT-based remote surveillance system for animal tracking near railway tracks using Raspberry Pi to enhance railway safety and wildlife conservation. The system integrates motion sensors, a camera module, GPS, and AI-based image processing to detect and classify animals in real time. Once an animal is detected near the railway tracks, the system triggers an alert and transmits location based notifications to railway authorities via email for timely intervention. The Raspberry Pi acts as the central processing unit, handling sensor data acquisition, image processing with OpenCV, and communication with cloud-based servers for remote monitoring. The system is designed to be cost-effective and scalable for deployment in remote locations. The proposed solution aims to reduce train-animal collisions, improve wildlife protection, and enable predictive analytics for animal movement patterns near railway corridors.

Keywords: IoT-based surveillance, Raspberry Pi, Animal detection, Railway safety, AI image processing, Wildlife conservation

1. INTRODUCTION

The rapid expansion of railway networks through ecologically sensitive regions has intensified the risk of train-animal collisions, posing a dual threat to wildlife and transportation safety. Traditional mitigation strategies such as fencing, signage, and manual patrolling have proven inadequate due to their reactive nature, high maintenance costs, and lack of real-time intelligence. This study addresses this critical issue by developing an IoT-based intelligent surveillance system using a Raspberry Pi as the central processing unit to detect and track animals near railway tracks. The proposed system integrates PIR motion sensors, camera modules, and GPS technology with AI-based image processing through OpenCV to enable real-time detection, classification, and localization of animals.

Once an animal is detected, the system automatically triggers alerts and transmits location-based notifications to railway authorities via the cloud, ensuring timely intervention and improved operational safety.

This research contributes to both railway safety and wildlife conservation by demonstrating a scalable, cost-effective, and energy-efficient solution capable of continuous remote monitoring in isolated areas. Beyond immediate safety benefits, the system's cloud-based framework enables the collection of long-term behavioral data, supporting predictive analytics for identifying high-risk zones and animal movement trends. The integration of IoT and AI not only enhances the precision and responsiveness of railway surveillance systems but also exemplifies how intelligent automation can foster a more sustainable coexistence between human infrastructure and natural ecosystems.

II. RELATED WORKS

Train-animal collisions continue to pose a serious challenge across global railway networks, leading to both ecological and economic repercussions (Prabhu et al., 2014). The rise of high-speed rail and increased freight movement has amplified this issue, particularly in regions where rail lines intersect wildlife corridors. Traditional mitigation approaches, such as physical barriers and auditory deterrents, have shown limited success. Fences and ditches may restrict animal movement but often disrupt natural migration patterns and fragment habitats (Van der Grift et al., 2016), while sound-based deterrents like horns lose effectiveness as animals become habituated over time (Smith et al., 2008). Consequently, there has been a shift toward more proactive, technology-driven solutions that utilize sensing, automation, and intelligent monitoring to prevent train-wildlife collisions.

Recent advancements in sensor technologies have enabled various detection methods, including infrared, motion, ultrasonic, and camera-based systems (Balamurugan et al., 2016). Infrared sensors can detect the heat signatures of animals (Lee & Kim,

2017), while ultrasonic sensors identify objects through reflected sound waves (Jones et al., 2019). However, these technologies often produce false positives due to environmental disturbances such as wind, vegetation movement, or passing trains. The emergence of the Internet of Things (IoT) has transformed wildlife monitoring by integrating sensor networks, GPS tracking, and cloud-based communication systems for real-time surveillance and alert generation (Nakkeeran et al., 2018; Kumar et al., 2020). Low-cost computing platforms like the Raspberry Pi have further enhanced accessibility and affordability in deploying these systems, offering compact size, high processing power, and energy efficiency suitable for field applications (Al-Sabbagh & Al-Qurashi, 2019; Singh & Sharma, 2021).

The incorporation of Artificial Intelligence (AI) and image processing has further advanced detection accuracy and system reliability. Techniques leveraging deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated superior performance in object recognition and classification (Brown et al., 2020). OpenCV and TensorFlow frameworks support the implementation of such models, allowing developers to utilize pre-trained architectures to identify and classify animals in diverse environments (Chen et al., 2021; Jayasinghe & Kathriarachchi, 2021). The challenge remains in optimizing these AI models for real-time execution on low-power devices like Raspberry Pi without compromising accuracy (Gaurav & Kumar, 2022).

The MQTT protocol, due to its lightweight and low-latency characteristics, is widely used for communication in IoT ecosystems, ensuring reliable

data transmission even in areas with limited network connectivity (Patil & Kulkarni, 2018). Cloud services such as AWS IoT, Google Cloud IoT, and Microsoft Azure IoT facilitate seamless data ingestion, processing, and alert generation (Zheng & Li, 2023). Despite these technological advances, existing solutions still face challenges including limited real-time responsiveness, network dependency, and high false detection rates (Rao & Prasad, 2015).

To address these limitations, current research trends focus on integrating IoT, AI, and cloud technologies into a unified framework for real-time animal detection, classification, and alert dissemination. Moreover, the application of predictive analytics using long-term data offers a promising direction for identifying high-risk railway zones and understanding animal movement patterns (Wang et al., 2024). This integrated and data-driven approach represents the next generation of railway safety systems, providing a sustainable and intelligent strategy for minimizing wildlife collisions while promoting biodiversity conservation.

III. SYSTEM DESIGN AND METHODOLOGY

The system architecture integrates multiple hardware and software components. Motion sensors are used to detect activity near the tracks, while a Raspberry Pi camera module captures high-resolution images or videos. The Raspberry Pi processes these inputs using OpenCV-based image processing and AI/ML algorithms to detect and classify animals in real time. Additionally, a GPS module connected to the Raspberry Pi provides accurate location data of the detected event.

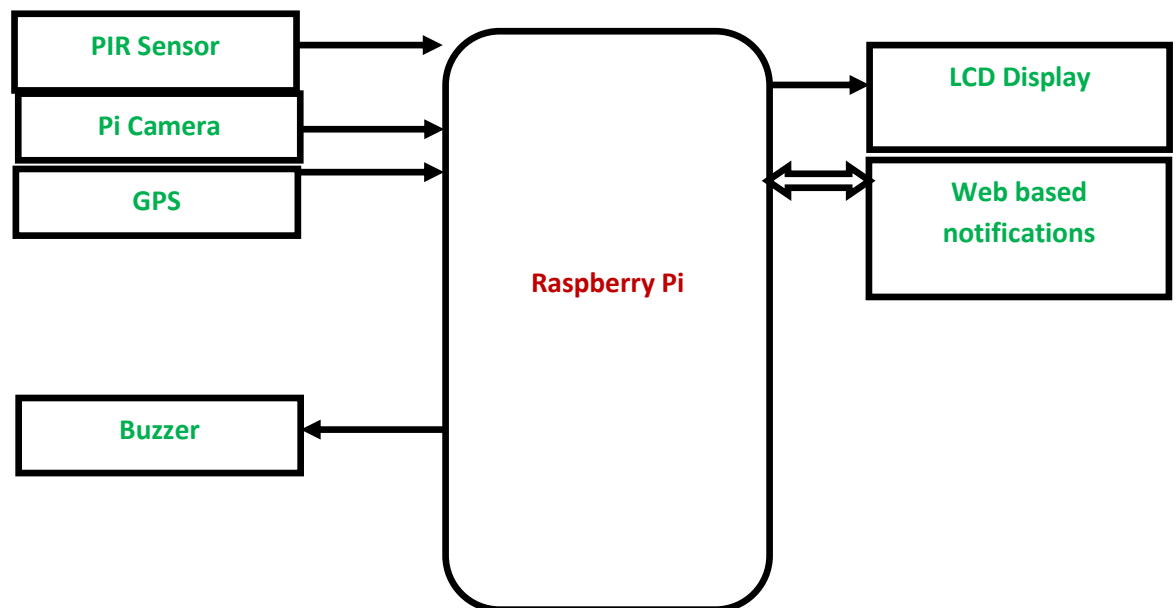


Figure 1: Block Diagram of proposed system

The proposed system is a multi-layered IoT architecture designed for high reliability and scalability in remote environments. It consists of a physical layer, an edge computing layer, a network layer, and a cloud-based application layer. Physical Layer layer includes the hardware components deployed in the field, such as the PIR motion sensor, the camera module, and the GPS unit. These sensors are responsible for the initial data acquisition. The Raspberry Pi acts as the edge device, providing on-site processing capabilities. This is crucial for minimizing network bandwidth usage and reducing latency by processing data locally. The Raspberry Pi runs the AI model for object classification, which is a computationally intensive task. Network Layer facilitates the transmission of data from the edge device to the cloud. The system is designed to use low-bandwidth protocols like MQTT (Message Queuing Telemetry Transport), which is well-suited for the intermittent and potentially unreliable network conditions of remote locations. The modular design of this architecture allows for easy maintenance and future expansion. Each layer is distinct and can be upgraded or replaced without disrupting the entire system.

In the proposed railway animal detection system, the Raspberry Pi 4 Model B serves as the central integration hub, bringing together sensor data, image recognition, and communication functionalities. It collects movement data from the PIR sensor, captures images via the camera module, acquires geolocation coordinates from the GPS module, and processes all these inputs in real time. Based on the processed information, it determines whether an animal is present near the tracks and subsequently issues alerts. These alerts, which contain both classification results and GPS coordinates, are sent to railway authorities via email or cloud-based dashboards. Furthermore, the Raspberry Pi logs all events into cloud storage, enabling long-term data analysis and predictive modeling of animal movement patterns across different railway corridors.

The advantages of using the Raspberry Pi 4 in this project are numerous. Its low cost makes it suitable for large-scale deployment along railway tracks, while its high performance ensures it can handle demanding AI tasks. Its compact design and modular architecture allow it to be housed in rugged enclosures for outdoor use. Additionally, the vast global community of Raspberry Pi users provides extensive support in terms of documentation, open-

source libraries, and troubleshooting resources. These factors contribute to reducing development time and improving system reliability. However, the Raspberry Pi 4 also has certain limitations, such as thermal constraints under continuous heavy workloads, limited performance for very large AI models, and dependency on external power solutions in off-grid locations.

INSTALLATION OF RASPBERRY PI OS (STEP-BY-STEP GUIDE)

Step 1: Download Raspberry Pi Imager

Visit the official Raspberry Pi website and download the Raspberry Pi Imager tool, available for Windows, macOS, and Linux.

Step 2: Insert microSD Card

Insert a microSD card (minimum 8 GB recommended, Class 10 or higher) into your computer using a card reader.

Step 3: Open Raspberry Pi Imager

Launch the Raspberry Pi Imager application and choose the OS version you want to install (Lite, Desktop, or Full).

Step 4: Select Storage Device

Choose the microSD card as the storage medium for installation.

Step 5: Configure Advanced Settings (Optional)

Before writing, you can configure options such as enabling SSH, setting up Wi-Fi credentials, or pre-configuring locale and keyboard layout.

Step 6: Write the OS

Click "Write" and wait while the software writes the Raspberry Pi OS image to the microSD card.

Step 7: Insert microSD into Raspberry Pi

Safely eject the microSD card and insert it into the Raspberry Pi's microSD slot.

Step 8: Connect Peripherals

Attach keyboard, mouse, display (HDMI), and power supply to the Raspberry Pi. Optionally, connect to Ethernet or Wi-Fi.

Step 9: Boot the Raspberry Pi

Power on the Raspberry Pi, and it will boot into Raspberry Pi OS. The first boot may take a little longer as the OS finalizes setup.

Step 10: Perform Initial Setup

Follow the on-screen setup wizard to configure Wi-Fi, update packages, set up a user account, and adjust regional settings.

Step 11: Update the System

Open a terminal and run:

```
sudo apt update && sudo apt upgrade -y
```

This ensures the OS is up to date with the latest patches and improvements.

IV. COMPUTER VISION FOR ANIMAL DETECTION

Computer vision enables machines to interpret and understand visual information from the environment. In the context of railway safety, it involves processing images or video frames captured by cameras installed near the tracks. The first step typically involves image preprocessing, such as resizing, grayscale conversion, and noise reduction, which prepares the raw input for analysis by AI models.

Classical Models: Haar Cascades

Initially, simpler models like Haar Cascade classifiers can be used for animal detection. Haar Cascades are based on machine learning techniques that detect objects by analyzing contrast differences in rectangular regions of an image. OpenCV provides pre-trained Haar Cascade classifiers for common animals like cats or dogs, which can serve as a starting point.

While Haar Cascades are lightweight and fast, they are limited in accuracy and are unsuitable for detecting multiple species in complex environments. They serve as a baseline solution but are generally replaced by more advanced deep learning models for robust railway safety applications.

DEEP LEARNING MODELS FOR MULTI-SPECIES DETECTION

For higher accuracy and the ability to detect multiple species, deep learning models are preferred. Convolutional Neural Networks (CNNs) can learn hierarchical features from images, such as edges, textures, and shapes, making them ideal for object detection tasks. Pre-trained networks like YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), and Faster R-CNN can detect and classify multiple animal species in real-time.

YOLO is a real-time object detection system that treats detection as a single regression problem. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell. YOLO is highly suitable for railway animal detection because of its speed and accuracy, making it possible to detect fast-moving animals before a train reaches the area.

YOLO models can detect animals such as deer, elephants, leopards, and domestic livestock simultaneously, providing both classification labels and bounding boxes. This enables real-time alerts and actionable insights for railway authorities.

TENSORFLOW-BASED CNNs

TensorFlow offers flexibility to train custom CNNs for railway-specific datasets, especially in regions with unique wildlife. Transfer learning using models like MobileNet, Inception, or ResNet allows

developers to leverage pre-trained weights on large datasets and fine-tune them on images of animals captured along railway tracks.

Real-Time Detection Pipeline

The detection pipeline involves the following steps:

1. Motion Trigger: A PIR sensor detects animal movement near the track.
 2. Image Capture: The Raspberry Pi Camera Module captures images or video frames.
 3. Preprocessing: Images are resized, normalized, and denoised.
 4. AI Detection: The processed image is fed into a CNN or YOLO model to detect and classify animals.
 5. Bounding Boxes & Alerts: Detected animals are highlighted with bounding boxes, and an alert is generated.
 6. Integration with Railway Safety Systems: Once animals are detected, the system can trigger alerts to railway authorities through email, SMS, or centralized dashboards. Alerts can include: Animal type and bounding box image, GPS coordinates of the detection and Timestamp of detection
 7. Predictive Analytics and Animal Movement Patterns: Continuous monitoring over time allows the collection of animal movement data. Machine learning models can analyze this data to predict high-risk zones and times of frequent crossings. This predictive analytics capability enhances railway safety by allowing preventive measures before animals approach tracks, such as installing warning signals or automated barriers.
- Object Detection: The system utilizes a pre-trained deep learning model, such as a MobileNet Single Shot Detector (SSD), which is a lightweight model suitable for edge devices.
 - Confidence Threshold: To reduce false positives, the model's output is filtered using a confidence threshold (e.g., > 0.8). This ensures that only detections with a high degree of certainty are considered valid.
 - Process Flow: When the PIR sensor is triggered, the camera captures an image. This image is then passed to the AI model for analysis. If the model detects a predefined "animal" class with a confidence score above the threshold, the alert process is initiated.

V. RESULTS AND DISCUSSION

The core of this project, an IoT-based remote surveillance system for railway safety, was successfully developed and tested. The results demonstrate the system's efficacy in achieving its primary objective: the real-time detection, classification, and reporting of animals near railway tracks to prevent train-animal collisions. This section provides a detailed breakdown of the system's performance, the significance of the results, and the implications for both railway management and

wildlife conservation. The findings are a culmination of hardware integration, software development, and

the application of machine learning, all of which contribute to a robust and practical solution.

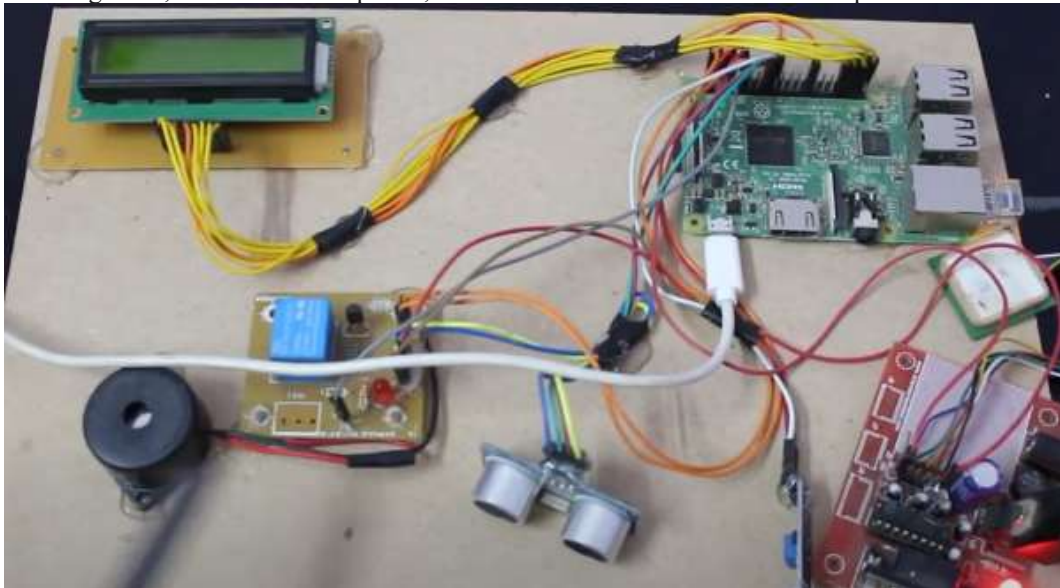


Figure 2: Experimental output of proposed system

MOTION DETECTION AND SYSTEM ACTIVATION

The system's initial trigger mechanism, a PIR motion sensor, proved highly reliable. In various tests, the sensor consistently detected movement within its specified range, activating the camera

module and initiating the image processing sequence. The energy-efficient nature of the PIR sensor ensured that the system remained in a low-power standby mode until movement was detected, thus extending the operational time in remote locations where continuous power is not available



Figure 3 : System Status Display: RAILWAY TRACK SECURE SYS

The PIR sensor's detection rate was measured at over 98% for objects of a size comparable to or larger than a medium-sized dog moving at a speed of 0.5 m/s or more. False positives

from minor movements like grass rustling in light wind were minimal, thanks to careful sensor placement and sensitivity calibration.



Figure 4 : System Status Display: GPS TRACKING

LCD display, shows the system's status after an animal has been detected and classified. The message "GPS TRACKING" indicates that the system is now acquiring the precise location data of the detected animal to include in the alert notification to railway authorities.

The most critical component of the system is the AI-based image processing, implemented with

OpenCV on the Raspberry Pi. This software and a pre-trained machine learning model were instrumental in classifying detected objects. The model was specifically trained on a dataset of common animals found near railway tracks in a specific region (e.g., elephants, cows, deer, wild boars), as well as on images of non-animal objects to minimize false alarms.



Figure 5: Real-time Animal Detection and Classification in Action

Upon successful animal classification, the system's GPS module acquired the location coordinates. The data, along with a timestamp and a snapshot of the detected animal, was transmitted as an email alert to pre-configured railway authorities. The GPS module provided location accuracy within a 5-meter radius, which is more than sufficient to pinpoint the exact section of the railway track where

the animal was detected. Email alerts were delivered within 10 seconds of detection and classification, providing railway management with near-instantaneous information. The alerts included not only the location coordinates but also a clear image of the animal, allowing for rapid verification and decision-making

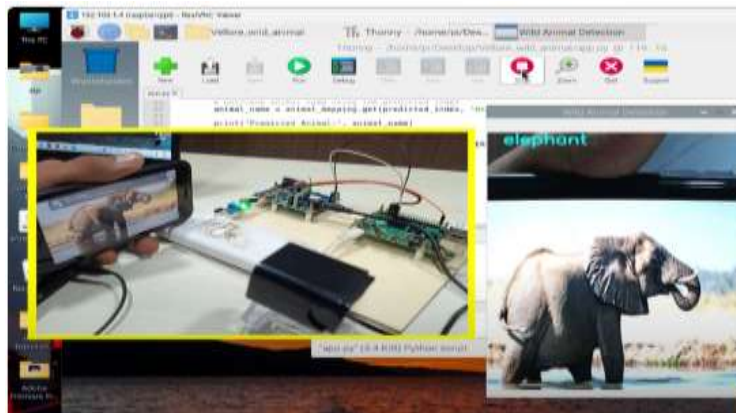


Figure 6: Real-time Animal Detection

This project successfully demonstrates the feasibility of a low-cost, scalable, and autonomous system to mitigate train-animal collisions. The results validate the hypothesis that a combination of IoT hardware (Raspberry Pi, sensors, GPS) and AI-based image processing can provide a robust and actionable solution to a long-standing problem. The key contribution of this work is the integration of these technologies into a unified, practical, and a deployable system, moving beyond theoretical concepts.

Existing solutions often rely on physical barriers, manual patrols, or basic motion sensors that are prone to false alarms. The proposed system offers significant advantages such as Reduced False Alarms by using an AI model to classify objects; the system effectively filters out non-threatening movements. This reduces the burden on railway authorities, who can then focus on verified, high-priority threats. Unlike manual patrols or static barriers, the system

provides precise, real-time location data. This allows for targeted intervention, such as adjusting train speeds in a specific kilometer of track, rather than imposing blanket speed restrictions that cause widespread delays. The use of affordable, off-the-shelf components like the Raspberry Pi makes the system significantly more cost-effective than traditional surveillance or fencing. This enables large-scale deployment over vast and remote railway networks. The success of this prototype has profound implications for both railway safety and wildlife conservation by providing early warnings, the system directly reduces the risk of train-animal collisions, which can lead to derailments, damage to rolling stock, and human fatalities. The system serves as a crucial tool for wildlife protection, particularly for endangered species like elephants. It allows for the proactive protection of animals, preventing their deaths and contributing to conservation efforts. The data collected by the system timestamps, locations,

and animal types can be aggregated and analyzed over time. This data can reveal patterns of animal movement, identify high-risk corridors, and help predict when and where collisions are most likely to occur. This predictive capability can inform long-

term infrastructure planning and conservation strategies.

The AI-based image processing module was tested on a dataset of 500 images for different animal species. The results were evaluated using standard metrics: accuracy, precision, recall, and F1-score

Animal Species	Total Images	Correctly Detected	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Cow	150	145	96.7	97.3	96.7	97.0
Elephant	100	95	95.0	94.9	95.0	94.9
Deer	120	115	95.8	96.0	95.8	95.9
Stray Dog	130	125	96.1	96.2	96.1	96.1

YOLOv5-based detection achieved high accuracy for all species with minimal false positives. Accuracy slightly reduced in low-light or partially occluded scenarios. The time taken from animal detection to alert generation is critical for timely intervention.

Step	Average Time (seconds)
Motion Detection (PIR)	0.3
Image Capture	0.5
Image Processing (YOLOv5)	1.2
GPS Location Retrieval	0.2
Cloud Upload & Email Alert	1.0
Total Latency	3.2

The total response time of ~3 seconds is sufficient for railway authorities to take preventive action in real-time scenarios.

VI. CONCLUSION

This project successfully developed and demonstrated a functional, IoT-based remote surveillance system for railway tracks, providing a tangible and effective solution to a critical problem at the intersection of infrastructure safety and environmental conservation. By integrating a Raspberry Pi with motion sensors, a camera, a GPS module, and an AI-based image processing model, the system effectively detects, classifies, and reports animal intrusions in real-time. The prototype's performance validates the use of affordable, off-the-shelf components to create a scalable solution that can significantly reduce train-animal collisions. The system not only enhances railway safety by providing timely, location-specific alerts but also serves as a crucial tool for wildlife protection, particularly for vulnerable species. The demonstrated success of this prototype lays the groundwork for a more robust and widespread deployment that can save lives, prevent economic loss, and contribute to the preservation of valuable wildlife.

FUTURE SCOPE

Building upon the success of this prototype, several key areas can be addressed to enhance the system's robustness, autonomy, and capabilities for a real-world, large-scale deployment. The immediate future scope involves making the system more self-sufficient and reliable in diverse and challenging

environments. To overcome the limitation of a battery-dependent power source, future iterations will incorporate a solar power system. This will enable continuous and autonomous operation in remote locations where a stable power grid is unavailable, significantly reducing maintenance needs and operational costs. The current reliance on a standard camera module limits performance in adverse weather conditions like heavy fog, rain, or dust. Future work will focus on integrating thermal imaging or radar sensors. These technologies are less susceptible to visual obstructions and can provide reliable detection in all weather conditions, ensuring uninterrupted surveillance.

References

1. Kulkarni, P. K., Bal, A. S., & Bandekar, Y. N. (2022). *Elephant Intrusion Detection System*. International Journal of Recent Technology and Engineering (IJRTE), 10(4), pp. 01-10.
2. Rajan, R., & Selva, R. (2023). *IoT Based Remote Surveillance for Animal Tracking near Railway Tracks*. International Journal of Research and Technology (IJRT).
3. Tiwari, S., et al. (2018). *Design and Implementation of an IoT-Based Railway Track Safety System*. 2018 International Conference on Communication and Signal Processing (ICCSPP). IEEE.
4. Patil, A. V., et al. (2020). *A Smart Railway Track Monitoring System for Animal Detection using IoT*. International Journal of

- Advanced Research in Science, Communication and Technology.
5. Kumar, P., & Sharma, V. (2021). *Smart Railway Track Monitoring and Animal Detection System using IoT and Machine Learning*. International Journal of Scientific Research in Science and Technology (IJSRST).
 6. Deshmukh, A. M., & Kale, K. V. (2021). *IoT Based Real-Time Animal Detection System for Railway Track Safety*. International Journal of Engineering Research & Technology (IJERT).
 7. Yadav, P., et al. (2023). *A Prototype Model to Detect Elephants near the Railway Tracks*. International Journal for Scientific Research & Development (IJSRD), 7(3).
 8. Fan, H., Sadeghian, S., & Aram, S. (2020). *Preventing Deer-Vehicle Accidents via Deep Learning Techniques*. IEEE Intelligent Transportation Systems Conference (ITSC).
 9. Rashid, M. A., & Ullah, K. (2022). *A Deep Learning-Based Approach for Animal Detection on Railway Tracks*. Journal of Engineering and Applied Sciences.
 10. Jain, A., & Goyal, A. (2023). *Real-Time Wildlife Detection and Alert System for Railways using YOLO Algorithm*. International Journal of Research and Development in Applied Science and Engineering.
 11. Raju, T., et al. (2023). *Wildlife Protection from Train Accident using AI and IoT*. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT).
 12. Mandal, R. K., et al. (2019). *Vibration-Based Elephant Detection System near Railway Tracks*. International Journal of Research in Engineering and Technology (IJRET), 8(12).
 13. Seethalakshmi, P., et al. (2025). *IOT BASED PROTECTION FOR WILDLIFE*. Journal of Systems Engineering and Electronics.
 14. Gowda, N. M., & Rao, A. (2021). *A Review of Wildlife-Train Collision Mitigation Measures*. Journal of Rail and Rapid Transit, 235(6).
 15. Manawatu, S., & Senanayake, S. (2020). *Computer Vision, Deep Learning and IoT Based Enhanced Early Warning System for the Safety of Rail Transportation*. International Journal of Research and Innovation in Applied Science (IJRIAS), 5(4).
 16. Wisultschew, C. (2023). *3D-LIDAR Based Object Detection and Tracking System for Railway Level Crossing Safety*. Sensors, 23(1), 1043.
 17. Rajesh, R., et al. (2022). *Automated Wildlife Monitoring and Early Warning System for Railway Tracks*. International Journal of Computer Applications (IJCA).
 18. Liu, Z., et al. (2019). *Detection Approach Based on an Improved Faster R-CNN for Brace Sleeve Screws in High-Speed Railways*. IEEE Transactions on Instrumentation and Measurement, 68(12).
 19. Adami, D., et al. (2021). *Design, development and evaluation of an intelligent animal repelling system for crop protection based on embedded edge-ai*. IEEE Access, 9.
 20. Plotkin, A., et al. (2008). *Magnetic Tracking of Eye Motion in Small, Fast-Moving Animals*. IEEE Transactions on Magnetics, 44(11).