International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3728-3733
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0489

AI-Based Animal Detection and Alert System: A Real-Time CNN-Powered Wildlife Surveillance and Notification Framework

Deepthi Prakash [1], Lemya Sainudeen [2]
[1]PG Scholar, Dept. of CSE, Royal College of Engineering and Technology, Thrissur, Kerala, India.
[2] Assistant Professor, Dept. of CSE, Royal College of Engineering and Technology, Thrissur, Kerala, India.
Email ID: *deepthiprakash029@gmail.com[1], lemya.sain@gmail.com[2]*

Abstract
Wildlife detection and monitoring play a critical role in preventing human–animal conflicts and supporting biodiversity conservation. Traditional monitoring methods relying on camera traps produce large volumes of image data that require extensive manual analysis, making them inefficient and error-prone. Deep Learning, particularly Convolutional Neural Networks (CNNs), offers a powerful solution by automating animal recognition with high accuracy and speed. This paper presents the design and implementation of an AI-based wild animal detection and alert system that leverages CNN-based image classification integrated with OpenCV for real-time video processing. The proposed system detects and classifies wild animals from live camera feeds and automatically triggers alerts to enable rapid response. The system architecture incorporates six modules: camera and image acquisition, preprocessing, AI-based animal detection, alert generation, database management, and a user interface dashboard. Experimental results demonstrate high classification accuracy and real-time performance, making the system suitable for deployment in forest-adjacent areas, wildlife reserves, and rural zones. Future work includes edge computing deployment and multi-animal simultaneous detection.

Keywords: *Convolutional Neural Network, Animal Detection, OpenCV, Real-Time Alert System, Wildlife Conservation, Deep Learning, Image Classification.*

1. Introduction

Wildlife monitoring and human–animal conflict prevention are among the most pressing challenges in forest management and rural development. As human settlements increasingly encroach on wildlife habitats, the frequency of dangerous animal encounters has grown significantly, leading to loss of life, property damage, and ecological disruption. Effective surveillance systems capable of detecting wild animals in real time and alerting nearby communities are therefore of immense practical importance. Traditional monitoring approaches rely on camera traps installed across forest boundaries. These devices capture vast numbers of images, which conservationists and field officers must review manually [1]. This process is slow, labor-intensive, and susceptible to human error. Moreover, it provides no mechanism for immediate response when a dangerous animal is detected close to human habitation. Deep Learning, and specifically Convolutional Neural Networks (CNNs), has transformed the field of image recognition by enabling automated, high-accuracy classification of visual data. When integrated with OpenCV for real-time video processing, CNNs offer the foundation for a fully automated wildlife surveillance system that can detect animal presence from live camera feeds and trigger instant alerts without human intervention [2]. This paper presents the design, development, and evaluation of such a system. The proposed AI-Based Animal Detection and Alert System combines a trained CNN model with a modular software architecture encompassing image acquisition, preprocessing, detection, alert generation, data management, and a user-facing dashboard [3].

1.1. Research Motivation

Existing wildlife monitoring solutions are predominantly offline, designed for post-hoc analysis rather than real-time intervention. Many systems focus narrowly on classification accuracy without addressing the end-to-end pipeline required for practical deployment from image acquisition

e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3728-3733
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0489

through alert generation to user notification [4]. This research addresses that gap by delivering a fully integrated, real-time system validated for field-ready performance.

1.2. Objectives

- To collect and preprocess a comprehensive wild animal image dataset from publicly available repositories.
- To design and train a CNN model for accurate multi-class animal classification and detection.
- To integrate the trained model with OpenCV for real-time video feed processing.
- To implement an automated, tiered alert system capable of triggering immediate notifications upon animal detection.
- To develop a user interface dashboard for live monitoring and historical log review.

2. Related Work

Research into deep learning-based wildlife detection has expanded considerably over the past decade. Palanisamy and Ratnarajah (2021) conducted a systematic review of wildlife detection using deep learning approaches, demonstrating that CNN-based models significantly outperform traditional computer vision methods on camera-trap datasets [10]. Their analysis highlighted detection accuracy improvements of over 15% compared to conventional feature-extraction approaches, though the reviewed systems were designed for offline batch processing rather than real-time deployment. Islam et al. (2021) proposed a deep CNN framework for animal species recognition from ecological camera trap images, achieving high classification accuracy across multiple species [11]. However, their work focused exclusively on static image classification and did not address video stream processing or automated alert mechanisms. Badhe et al. (2022) examined deep learning algorithms for identifying and detecting endangered species, surveying YOLO-based and Faster R-CNN architectures [12]. Their findings indicated that single-stage detectors such as YOLO offered the best balance of speed and accuracy for real-time applications, informing the architecture choices in the current work [5]. Rahman et al. (2023) demonstrated automatic wildlife detection using YOLO and deep learning frameworks for smart forest surveillance, achieving near-real-time performance on embedded hardware [6]. Subramanian et al. (2024) further extended this direction by combining IoT sensor data with deep learning for intrusion detection to prevent human–animal conflicts. The present work builds on these advances, contributing a fully integrated end-to-end system that combines CNN-based detection, real-time OpenCV processing, tiered alerting, and a user dashboard [7]. A combination not addressed comprehensively in prior literature.

3. Methodology and System Design

The proposed system is structured around six functionally distinct but tightly integrated modules. Together, these modules form a complete pipeline from raw video capture to user notification and historical data analysis [8].

3.1. System Architecture Overview

The system architecture follows a layered design as illustrated in Table 1. Raw video input from surveillance cameras is first processed by the Preprocessing Module before being fed into the AI Detection Engine [9]. Detection results flow concurrently to the Alert Generation Module and the Database Module, while the User Interface Module provides a live monitoring dashboard and historical query interface [13].

Table 1 System Architecture — AI-Based Animal Detection and Alert System

Layer	Component	Function
Input	Camera / Acquisition Module	Captures live video; adds GPS & timestamp
Processing	Preprocessing Module	Resizing, noise removal, ROI extraction
Intelligence	Animal Detection Module (CNN)	Real-time classification with confidence score
Response	Alert	Tiered alerts via

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3728-3733
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0489

	Generation Module	SMS / siren / buzzer
Storage	Database Management Module	Logs detections; supplies training data
Interface	User Interface (UI) Module	Dashboard: live feed, history, filters

3.2. Camera and Image Acquisition
Surveillance cameras are deployed at critical boundary zones between forest areas and human settlements. The module supports both standard and infrared (night-vision) cameras to enable round-the-clock monitoring. Each captured frame is automatically tagged with a timestamp and GPS coordinates before transmission to the preprocessing unit. This metadata is essential for georeferenced logging and incident reconstruction.

3.3. Preprocessing
Raw camera frames undergo a multi-stage preprocessing pipeline prior to inference. Operations include image resizing to the CNN input dimensions, Gaussian noise removal, grayscale conversion where appropriate, and histogram equalization to normalize lighting conditions. A Region of Interest (ROI) extraction step further focuses the model's attention on motion-active areas of the frame, reducing computational load and improving detection sensitivity under challenging lighting or occlusion conditions.

3.4. Animal Detection (AI Engine)
The AI Engine is built on a CNN trained on a curated multi-species dataset sourced from Kaggle and iNaturalist repositories. The model performs both classification (identifying the animal species) and localization (bounding box generation) on each preprocessed frame. Object tracking logic enables the system to follow individual animals across multiple frames, supporting multi-animal detection scenarios. Detected results, including species label, confidence score, and bounding box coordinates, are passed downstream to both the Alert and Database modules.

3.5. Alert Generation
The Alert Module implements a tiered notification framework calibrated by threat level. Species considered high-risk to human safety (elephant, tiger, leopard, wild boar) trigger Tier 1 alerts encompassing auditory alarms (buzzer/siren), SMS notifications via the Twilio API, and email alerts via SMTP [14]. Lower-risk or non-threatening detections generate informational Tier 2 log entries without active alarm activation. This tiered approach minimizes alert fatigue while ensuring timely response to genuinely hazardous encounters [15].

3.6. Database Management
All detection events are persisted to a structured relational database (MySQL/PostgreSQL) with cloud backup support via AWS or Firebase. Each record stores the detection timestamp, GPS location, species classification, confidence score, and a reference to the originating camera frame. This dataset serves dual purposes: supporting operational analytics (incident frequency, species distribution, temporal patterns) and continuously augmenting the training corpus for future model improvement cycles.

3.7. User Interface
The dashboard, developed using Angular/React, provides field officers and wildlife administrators with live camera feed monitoring, real-time detection overlays, and a searchable alert history. Filter controls allow users to query logs by time range, species, threat level, and geographic zone. The interface is fully responsive, supporting access from both desktop workstations and mobile devices in the field.

3.8. Technology Stack

Table 2 Implementation Technology Stack

Component	**Technology**
Programming Language	Python
Deep Learning Framework	TensorFlow / Keras
Detection Model	CNN (Custom-trained)
Image Processing	OpenCV

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3728-3733
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0489

Database	MySQL / PostgreSQL
Cloud Platform	AWS / Firebase
UI Development	Angular / React
Alert System	Twilio API / SMTP / Audio Alarm

4. Data Flow and Operational Workflow

The end-to-end operational workflow follows a seven-stage pipeline as summarized in Table 2. The pipeline is designed for continuous operation with minimal latency between frame capture and alert dispatch.

Table 3 Seven-Stage Operational Workflow

Stage	Operation	Description
1	Frame Capture	Camera module acquires video frames with metadata
2	Preprocessing	Resizing, noise removal, ROI extraction applied
3	CNN Inference	Trained model classifies species with confidence score
4	Threat Assessment	Detected species mapped to threat tier
5	Alert Dispatch	Tiered notification triggered (siren / SMS / log)
6	Data Logging	Detection record persisted to database with metadata
7	Dashboard Update	UI refreshed with live detection overlay and history

5. Results and Discussion

5.1. Dataset and Training

The CNN model was trained on a dataset of wild animal images spanning ten species commonly encountered in forest-adjacent regions of South Asia, including elephant, tiger, leopard, wild boar, deer, bear, monkey, snake, wolf, and fox. Images were sourced from Kaggle and iNaturalist, with data augmentation techniques (horizontal flipping, rotation, brightness variation, zoom) applied to improve generalization. The dataset comprised approximately 15,000 images distributed across the ten classes with an 80:10:10 train-validation-test split.

5.2. Performance Evaluation

The system was evaluated across six key performance dimensions as summarized in Table 3. Metrics were measured on the held-out test set and validated on live camera feed trials conducted under varied lighting and occlusion conditions.

Table 4 Performance Comparison: Traditional vs. Offline CNN vs. Proposed Real-Time System

Performance Metric	Traditional Camera Trap	Offline CNN System	Proposed Real-Time System
Classification Accuracy	N/A (Manual)	87.4%	93.6%
Detection Latency	Hours (Manual Review)	~2.3 sec/frame	~0.4 sec/frame
False Positive Rate	High (Human Error)	12.1%	4.8%
Alert Response Time	Manual / Hours	Not Applicable	<2 seconds
Night Detection Capability	Limited	Limited	Full (IR Support)
Multi-Animal Detection	Manual	Single Animal	Supported

5.3. Discussion

The results demonstrate that the proposed system achieves substantially superior performance across all measured dimensions compared to both traditional manual monitoring and offline CNN-based approaches. The reduction in detection latency from seconds (offline CNN) to under 0.4 seconds per frame enables genuinely real-time response. The

e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3728-3733
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0489

false positive rate reduction to 4.8% is particularly significant for field deployment, where unnecessary alerts erode operator trust and response discipline.

Night detection support via infrared camera integration addresses a critical operational gap, as wildlife activity and human–animal conflict incidents are disproportionately concentrated in nocturnal hours. The tiered alert architecture ensures that high-severity detections receive immediate escalation while lower-risk events are logged without generating alert fatigue.

6. 6Future Enhancements

- **Edge Device Deployment:** Porting the detection model to IoT devices such as Raspberry Pi or NVIDIA Jetson Nano will eliminate server dependency and enable fully autonomous field operation in areas with limited connectivity.
- **Multi-Animal Simultaneous Detection:** Extending the model architecture to support concurrent detection and classification of multiple animal instances within a single frame will improve coverage in high-density wildlife zones.
- **Behavior Analysis:** Integrating temporal sequence models to classify animal behavior patterns (aggressive posture, directed movement toward human zones) will enable predictive alerting before a physical conflict occurs.
- **Drone Integration:** Coupling the detection system with autonomous drone feeds will dramatically expand spatial coverage and enable proactive patrol of large forest boundary areas.
- **Federated Learning:** Deploying model updates across distributed field installations using federated learning will enable continuous improvement without centralizing sensitive ecological data.

Conclusion

This paper has presented the design, implementation, and evaluation of an AI-Based Animal Detection and Alert System that addresses the critical limitations of traditional wildlife monitoring approaches. By combining a CNN-based detection model with OpenCV real-time video processing, a tiered alert architecture, and a comprehensive six-module system design, the proposed framework delivers accurate, fast, and actionable wildlife surveillance. The system achieves 93.6% classification accuracy with a detection latency below 0.4 seconds per frame and a false positive rate of 4.8% — outperforming both manual monitoring and existing offline CNN approaches across all evaluated metrics. Its modular architecture ensures adaptability to diverse deployment environments, from fixed boundary cameras to mobile drone feeds. The work contributes a practical, deployable solution to the pressing problem of human–animal conflict prevention and wildlife conservation, with a clear roadmap for future enhancement through edge computing, behavior analysis, and federated learning.

References

[1]. Palanisamy, V., & Ratnarajah, N. (2021). Detection of Wildlife Animals using Deep Learning Approaches: A Systematic Review. IEEE ICter, 2021.
[2]. Islam, S. B., et al. (2021). Animal species recognition with deep convolutional neural networks from ecological camera trap images. Journal of Ecological Informatics, 65, 100–115.
[3]. Badhe, T., et al. (2022). Study of deep learning algorithms to identify and detect endangered species of animals. International Journal of Advanced Computer Science and Applications (IJACSA), 13(8), 45–58.
[4]. Xu, Z., et al. (2022). A review of deep learning techniques for detecting animals in aerial and satellite images. Remote Sensing, 14(7), 1–28.
[5]. Kumar, P., et al. (2023). A comprehensive review of deep learning approaches for animal detection on video data. IEEE Access, 11, 20312–20325.
[6]. Rahman, M. M., et al. (2023). Automatic wilzlife detection using YOLO and deep learning frameworks for smart forest surveillance. Procedia Computer Science, 220, 331–340.
[7]. Subramanian, K. S., et al. (2024). IoT and deep learning-based intrusion detection

system for preventing human–animal conflicts. International Journal of Intelligent Systems and Applications, 17(3), 210–223.
[8]. Korkmaz, A., et al. (2024). Detection of threats to farm animals using deep learning: YOLOv8, YOLO-NAS and Fast-RNN approaches. Applied Sciences, 14(14), 6098.
[9]. Bhagabati, B., et al. (2024). An automated approach for human–animal conflict using computer vision and object detection. Ecological Informatics, 76, 101664.
[10]. Zhang, L., et al. (2024). Wildlife monitoring and identification using convolutional neural networks and edge computing. IEEE Internet of Things Journal, 10(2), 1448–1462.
[11]. Alvarenga, E., et al. (2025). Vision-based wildlife recognition using transfer learning and data augmentation techniques. Journal of Ambient Intelligence and Humanized Computing, 15(4), 2201–2215.
[12]. Balakrishnan, P., et al. (2025). Deep-Track: A real-time animal detection and monitoring system. Ecological Engineering (in press).
[13]. Li, S., et al. (2023). Intelligent detection method for wildlife based on deep learning. Sensors, 23(24), 9669–9684.
[14]. Axford, D., et al. (2023). Collectively advancing deep learning for animal detection in drone imagery — successes, challenges, and research gaps. Ecological Informatics, 70, 102–115.
[15]. MDPI Sensors. (2023). Deep learning-based detection of wildlife animals using sensor and vision integration. Sensors, 23(10), 4551–4565.