International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

IoT-Based Real-Time Wild Animal Detection and Alert System Using Deep Learning and Multi-Sensor Integration

Kolluru Likhitha[1], Bondalapati Reshma Chowdary[2], Chilakanti Praneeth[3], Kammula Sree Satya[4], K. Jairam[5]
[1,2,3,4,5] Computer Science and Engineering, SRK Institute of Technology, Vijayawada, India
Email ID: *likhithakolluru28@gmail.com[1], chilakantipraneeth@gmail.com[2], reshmabondalapati28@gmail.com[3], sreesatyakammula@gmail.com[4], ramjai4@gmail.com[5]*

Abstract
Throughout the world - especially those areas close to Forest Ecosystems - humans often face difficulties or challenges created as a result of conflict with various types of animals around them. Such conflicts arise when livestock have been attacked by wild animals, or when there is a direct threat to the safety of humans from an attacking wild animal(s) that may cause harm or even death. This article introduces a new IoT based intelligent wildlife monitor/detection system. It will use the YOLOv8 deep learning algorithm/architecture combined with Environmental Sensor Networks (ESNs), as well as ESP32 microcontrollers for Real-time Data Acquisition (RDA) from 3 types of sensors: DHT11 Temperature & Humidity Sensor, MQ135 Air Quality Sensor and HC-SR04 Ultrasonic Distance Sensor; and provide the user with a way to monitor wildlife movement through actual real-time webcam images using access control (RBAC) based website (Flask Framework), receive alerts about detected wildlife on multiple channels (Telegram API, Google Text-to-Speech audio notifications and LCD displays) and maintain historical records of previous detection events and all user accounts through an SQLite Database. By testing the prototype and analyzing its findings against the traditional methods used; it offers a much better response time to classify detected animals (under 2 seconds) and provides a much longer period of time that the user has to respond than traditional methods of monitoring/watching for wildlife attacks.

Keywords: *Wildlife Detection, YOLOv8, IoT, ESP32, Deep Learning, Human-Wildlife Conflict, Computer Vision, Multi-Sensor Integration*

1. Introduction
The increase in human settlement encroaching upon wildlife habitat has increased the number of human-animal conflicts occurring, especially for agricultural communities who live close to forests [1]. All traditional methods for deterring wildlife (e.g., barriers, manual monitoring, deterrent sounds) have the following limitations: They require too much labour; response time is longer than 1 hour; and the false positive rate is too high. However, recent developments in computer vision, especially through deep learning-based object detection, have opened up new possibilities of improving automated intelligent monitoring systems using low-cost IoT hardware [2][3]. This paper discusses the results of developing an integrated IoT-based wildlife detection system that uses the YOLOv8 object detection algorithm combined with multiple sensors to monitor the environment surrounding wildlife. The integrated IoT-based system allows real-time classification of animals, context classification of the environment, and the dissemination of alerts to users via web-based, telecommunication, and on-site auditory/visual notification. The deployment of role-based authentication framework permits both administrative oversight of devices, as well as facilitation of forest rangers in the field. The key contributions of the paper are: 1) integrating the advanced YOLOv8 object detection algorithm with IoT sensor networks for contextual threat assessment; 2) development of a distributed edge-cloud model to support real-time processing; 3) development of a multi-modal alerting system that provides users with visual, auditory, and telecommunication means of alert; 4) deployment of a secure web-based

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

monitoring platform with role-based access control; 5) persistent storage of the database for analysis of historical trends.

2. Related Work

2.1.Deep Learning for Wildlife Detection

Significant progress has been made in image classification due to the use of convolutional neural networks (CNNs) with the development of VGGNet architecture by simonyan and zisserman which led to significant improvements in object classification [6]. Further development of CNNs for very deep networks was achieved by residual connections in resnet networks ([8]). Using a different approach YOLO or you only look once has revolutionized the field of real time object detection by making detection a regression problem ([2]). Current methods of using YOLOV5 to detect wildlife have produced promising results ([9][10]). Archana et al developed an artificial intelligence based wild animal detection system with an artificial neural network which shows that automated systems of this type are feasible ([4]). Gourisaria et al researched a way to use the deep learning method to classify animal species ([5]). Research on camera trap data processing has demonstrated that deep learning methods can be used for ecological monitoring ([10][13][14]).

2.2.IoT-Based Wildlife Monitoring

There has been an increasing interest in the usage of Internet of Things (IoT) technologies within the realm of wildlife monitoring. Saxena et al. (2018) have developed a deep learning-based animal detection and accident prevention system, while Patil and Ansari (2017) designed an intrusion detection solution for wild animals utilizing artificial intelligence of things. Thermal imagery and Generative Adversarial Networks (GANs) for animal detection within the realm of surveillance applications were explored by Khatri et al. (2020) .

2.3.Multi-Sensor Integration

Combining environmental sensors with visual detection enhances the accuracy of detecting animals by providing contextually relevant information. Research suggests that the combination of visual detection and environmental parameters leads to greater accuracy of threat-level assessments and lower rates of false positives within automated monitoring systems.

3. System Architecture

3.1.Overall System Design

The architecture for the proposed system has four main components: edge sensing, which uses an ESP32 microcontroller with built-in sensors; computer vision processing, which uses YOLOv8 to detect objects via camera; communication and alerting, which uses multiple channels (i.e., email, text, etc.) to send out notifications; and a web application layer that provides a user interface and stores data. A visual representation of the architecture can be seen in Figure 1.

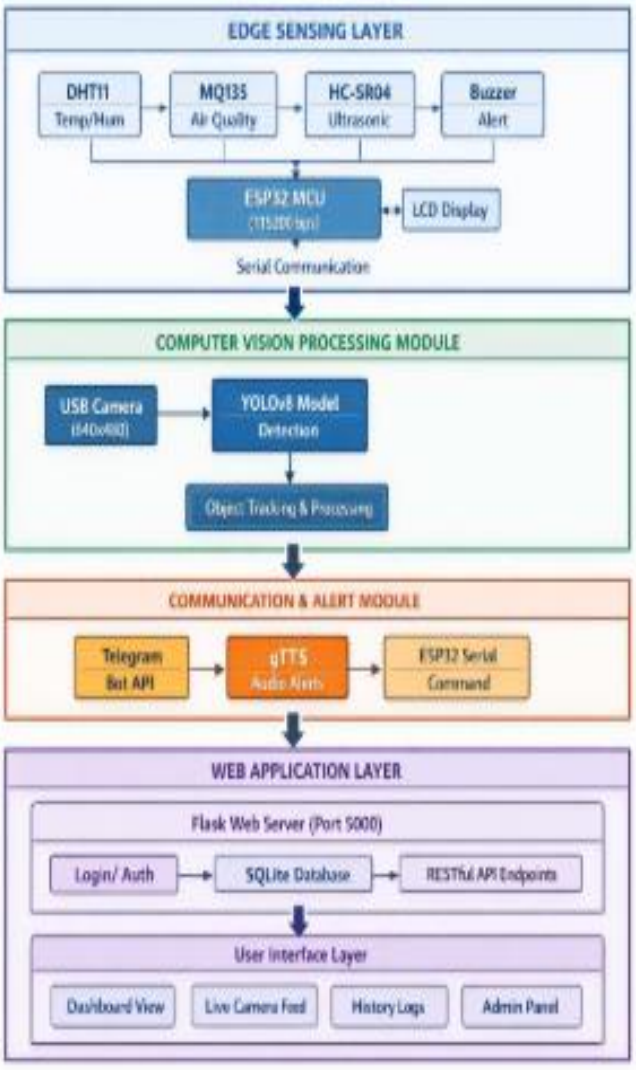

Figure 1 System Architecture Block Diagram

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

3.2.Hardware Components and Specifications

Table 1 System Hardware Components and Specifications

Component	Specifications	Function
ESP32 Microcontroller	115200 baud, Serial UART	Edge computing and sensor data acquisition
DHT11 Sensor	Temperature: 0–50°C Humidity: 20–90% RH Sampling: 1 Hz	Environmental monitoring
MQ135 Sensor	ADC: 0–4095 (12-bit) Quality Index: 0–100	Air quality measurement
HC-SR04 Sensor	Range: 0–4 m, Ultrasonic	Proximity detection for wildlife monitoring
16×2 LCD Display	I2C (0x27), 32 characters	Local data visualization
Active Buzzer	Programmable patterns	Audible alert generation

3.3.Software Architecture and Specifications

Table 2 Software Architecture and Specifications

Component	Specifications	Function
YOLOv8n Detection Engine	Model: YOLOv8 nano variant Resolution: 640×480 pixels Frame Rate: 30 FPS Species: 18 wildlife classes Confidence Threshold: 0.5	Real-time wildlife detection and classification
Flask Web Framework	Architecture: RESTful API Processing: Multi-threaded Endpoints: Data streaming, historical records, admin operations	Concurrent sensor acquisition, video processing, and web services
SQLite Database	Tables: Users, Detections, Alerts Security: Foreign key constraints Indexing: User sessions and events	Persistent data storage with relational integrity
Authentication System	Hashing: PBKDF2-SHA256 (Werkzeug) Access Control: Decorator-based Roles: Admin and Field Officer	Secure user authentication and role-based authorization

4. Implementation Details

4.1.Edge Device Firmware

An implementation of a polling architecture that uses a 1-second sampling interval, implemented in the ESP32 firmware. In order to determine the accuracy of the measurements taken by these sensors, the values measured for temperature and humidity are validated for NaN conditions, while air quality measurements made by the sensor have their corresponding 12-bit ADC values mapped onto a percentage scale. Ultrasonic distances that exceed

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

400cm are determined to be invalid measurements and filtered out. The ESP32 uses a comma-delimited format to provide serial output data to the host system, with the format for the serial data being "SENSOR_DATA, temp, humidity, airquality, distance". This comma-delimited format makes parsing data on the host much easier. The firmware uses a buzzer that generates 5 pulses at 200ms intervals when an "ALERT: Animal Name" command has been sent via the serial connection. Additionally, the firmware updates the LCD display to indicate that an alert has been triggered by displaying "ALERT!" along with the species name, and it remains in an alert state until it has been sent a "STOP_ALERT" command.

4.2.Computer Vision Pipeline

The detection pipeline uses video frames for processing, which consist of the following steps:

- Frame Acquisition: The OpenCV Video Capture interface captures frames from a USB camera configured to capture video at a resolution of 640 × 480 pixels with a frame rate of 30 frames per second (FPS)
- Object Detection: The YOLOv8 model detects objects on each frame that pass the target identification threshold of 0.5. For every object detected, the object detection module returns bounding box coordinates, class labels, and confidence scores
- Classification Filtering: Detected objects are filtered against a pre-established list of wild species in the deployment area to eliminate all detections that are not wildlife related
- Temporal Filtering: The system uses a 10-second cooldown to prevent users from receiving excessive alerts for the same species within a certain period of time and to allow the system to respond to additional sightings of the same species without having to provide immediate alerts
- Visualization: Boxed object detections (red boxes) are labelled with the name of the species and a confidence score based on the analysed video. System and distance of the detected object overlays are also shown on every video frame to give the end user a complete picture of what is occurring around them.

4.3.Alert Dissemination System

- Notifications sent to the Telegram Messaging app via bots whenever an animal's presence was detected after the cooldown time has ended. Notifications would contain formatted message text denoting the type of animal, when it was detected, and how far the animal is from you, then sent using POST requests to a Telegram messaging app.
- Audio messages created using the Google Text To Speech library, with audio message formats based on urgency levels that correlate to the distance of the detected animal in relation to the user. Under 50cm is "critical"; 50cm - 100cm is "high alert"; 100cm + is "warning." These audio clips are played through the pygame mixer, and saved to conserve space.
- A detection event will send a serial command to the ESP32 chip to trigger the buzzer and LCD screen to give immediate notification of the detection event at that location.

4.4.Web Application Features

A dashboard interface provides a visual representation of sensor data (temperature, humidity, air quality, and distance to create dynamic progress bars) and displays the most recent wildlife detection in a panel with a timestamp and distance from the user. Data displayed within the dashboard updates once per second through AJAX polling of the REST API endpoints. Using an MJPEG video stream, clients can view live video feeds of processed video along with annotations indicating what animals have been detected. Live indicators are overlaid on the video and show detection notifications based on the surrounding environment and other status information about the detection system. Interactive controls are available to allow users to refresh their feed, take snapshots, view in full screen mode or mute alerts. A detection history log search tool allows users to search through historical drama records and view on a timeline with color coded alerts indicating severity (red for dangerous species, yellow for moderate threat, and green for no threat). Each

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

historical entry contains complete metadata, including species name, detection timestamp, detection distance, confidence rating, and environmental conditions at the time of the detection. Metrics aggregated in an 'administrative' dashboard allow users to analyse total number of registered users, total number of alerts, number of active and inactive wildlife threats, total number of notified wildlife encounters, among other stats. A tabular view also shows the most recent user registrations and alert histories, including their corresponding active and inactive status. In the case of an emergency, there is an emergency contact page that allows for quick access to the three primary emergency services (forest officers, fire department, and ambulance). Each also includes the ability to make a phone call with one click and specifies the hours of availability (e.g., 24/7 available).

5. Experimental Setup and Results

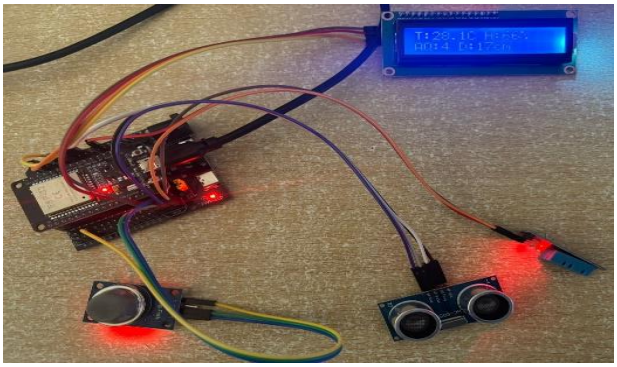

Figure 2 Hardware Implementation

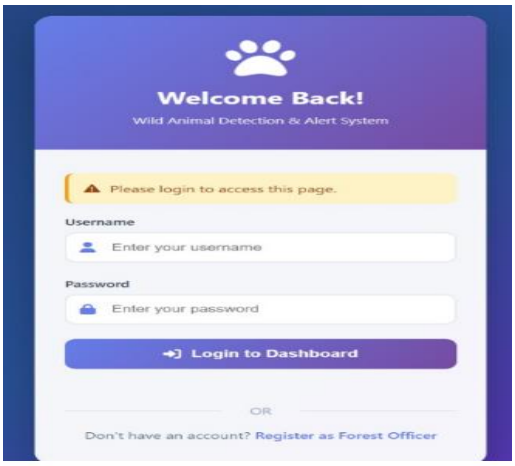

Figure 3 Forest Officer Login Page

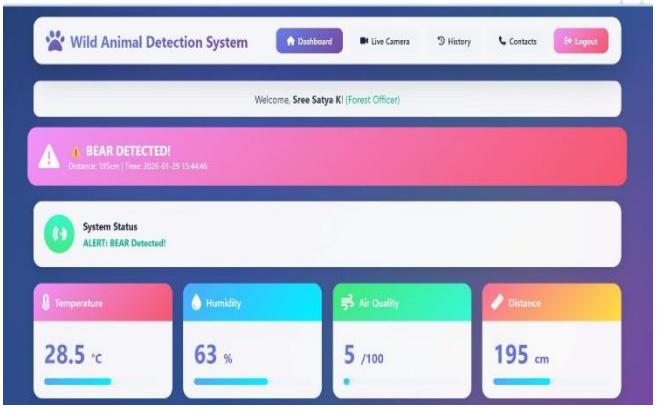

Figure 4 Dashboard

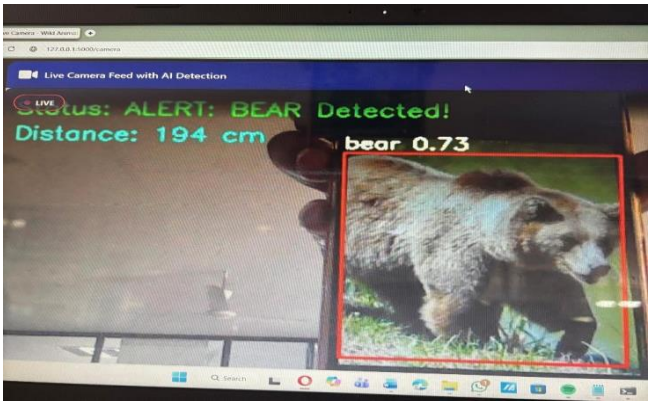

Figure 5 Live Camera

Figure 6 Aletrs to the telegram chatbot

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1144-1150
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0162

Conclusion

A comprehensive IoT wildlife detection system that incorporates YOLOv8 deep-learning architecture and multiple sensor environmental monitoring has been established in this paper. This system successfully demonstrates real-time classification of animals within less than 2 seconds and utilizes multiple alert mechanisms to notify users when an animal has been detected. In addition, the web interface provides role-based access control (RBAC) for end users, which provides an effective framework for implementing these systems in forestry-related management operations. The results from the testing phase of this device validate the feasibility of using this system to mitigate human-wildlife conflict associated with agricultural areas that border forested areas. The modular design of this system allows for future improvements in the form of multi-camera support, training of specific machine learning models to accommodate the local wildlife population in a particular region and implementing advanced analysis methods to determine the behavioral patterns of detected animals. The use of these devices will reduce agricultural crop loss and reduce livestock deaths while encouraging wildlife and human populations to co-exist near agricultural operations.

References

[1]. "Man-Animal Conflict Rose in Gir After the Population Increase of Lions in Gir, Jagat," Available: https://shorturl.at/DLN08 (Accessed: 30.08.2023)

[2]. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.

[3]. M. J. Wilber, W. J. Scheirer, P. Lecher, T. S. Heflin, J. Zott, and T. E. nBoult, "Animal recognition in the mojave desert: Vision tools for field biologists," in 2013 IEEE Workshop on Applications of Computer Vision (WACV), 2013, pp. 206-213.

[4]. H. Archana, H. Poornima, K. Raksha, V. Shwetha, and H. R. Supreeth, "Artificial Neural Network Based Image Processing for Wild Animal Detection and Monitoring," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, no. 5, 2017.

[5]. M. K. Gourisaria, M. K. Singh, R. Jha, A. Yadav, and P. Koedam, "Performance Enhancement of Animal Species Classification Using Deep Learning," in International Conference on Computing, Communication and Learning, Cham: Springer Nature Switzerland, pp. 208-219.

[6]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[7]. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in International Conference on Machine Learning, PMLR, 2019, pp. 6105-6114.

[8]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[9]. N. Mamat, M. F. Othman, and F. Yakub, "Animal Intrusion Detection in Farming Area using YOLOv5 Approach," in 2022 International Conference on Computer and Drone Applications (IConDA), 2022, pp. 1-5, IEEE.

[10]. J. Schneider, G. W. Taylor, and S. Kremer, "Deep learning object detection methods for ecological camera trap data," in 2018 15th Conference on Computer and Robot Vision (CRV), 2018, pp. 321-328, IEEE.

[11]. A. Saxena, A. Goyal, and S. Singh, "An animal detection and collision avoidance system using deep learning," in Advances in Communication and Computational Technology: Select Proceedings of ICACCT 2019, 2021, pp. 1069-1084, Springer Singapore.

[12]. Y. Zhu, T. H. Vu, and I. W. H. G. Tan, "Towards automatic wild-animal detection in low-quality camera-trap images using two-channeled perceiving residual pyramid

networks," in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2860-2864.

[13]. M. S. Norouzzadeh et al., "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," Proceedings of the National Academy of Sciences, vol. 115, no. 25, pp. E5716-E5725, 2018.

[14]. A. Gomez, G. D. Perez, A. A. Salazar, and A. Diaz, "Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds," in International Symposium on Visual Computing, Cham: Springer International Publishing, 2016, pp. 747-756.

[15]. H. D. Patil and N. F. Ansari, "Intrusion detection and repellent system for wild animals using artificial intelligence of things," in 2022 International Conference on Computing, Communication and Power Technology (IC3P), 2022, pp. 291-296, IEEE.

[16]. K. Khatri, A. Asha C. S., and J. M. D'Souza, "Detection of animals in thermal imagery for surveillance using GAN and object detection," in 2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), 2022, pp. 1-6, IEEE.

[17]. A. Shukla and S. Anand, "Metric learning based automatic segmentation of patterned species," in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3982-3986, IEEE.

[18]. D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A deep learning approach for face detection using YOLO," in 2018 IEEE Punecon, 2018, pp. 1-4, IEEE.

[19]. K. Daveand S. Bhatia, P. Goel, A. Ganatra, and K. Kotecha, "An amalgamation of YOLOv4 and XGBoost for next-gen smart traffic management system," PeerJ Computer Science, vol. 7, p. e586, 2021.

[20]. "Animals Detection Images Dataset," Kaggle. Available: https://kaggle.com/datasets/antoreepjana/animals-detection-images-dataset (Accessed: 29.07.2023)

[21]. "NTLNP Dataset," Available: https://huggingface.co/datasets/myyyxy/NTLNP (Accessed: 02.08.2023)

[22]. "Alpha Make Sense," Available: https://www.makesense.ai/ (Accessed: 02.08.2023)

[23]. "Albumentations: Albumentations is a computer vision tool that boosts the performance of deep convolutional neural networks," Available: https://albumentations.ai/ (Accessed: 15.08.2023)

[24]. "Ultralytics," Available: https://www. ultralytics. com/ (Accessed: 04.08.2023)