ISSN2277-2685
IJESR/April-June 2026/ Vol-16/Issue-2/766-775
G.Sai Ganesh *et.al.,*/International Journal of Engineering & Science Research

Full Length Article

IoT Based Real-Time Animal Detection System For Smart Agricultural Crop Protection

G.SaiGanesh[1], P.Ushasri[2], S.Vishal[3], Y.Srinija[4], Dr.Md.ASIF[5]
[1,2,3,4]B.Tech Student's; Department Of Electronics And Computer Engineering
J.B. Institute Of Engineering And Technology, Hyderabad, India.
[5]Associate Professor; Department Of Electronics And Computer Engineering
J.B. Institute Of Engineering And Technology, Hyderabad, India.

Accepted 26-04-2026

Abstract

Human–wildlife conflict has intensified in recent years due to deforestation, agricultural expansion, and habitat fragmentation, forcing wild animals to enter human settlements and farmlands. This study proposes a cost-effective and automated animal intrusion detection system that leverages an ESP32-CAM module integrated with Internet of Things (IoT) technology and deep learning algorithms.The system continuously captures real-time images and processes them using a trained object detection model to identify the presence of wild animals. Upon detection, instant alerts are transmitted to users such as farmers or forest authorities via mobile applications, enabling timely preventive action. Additionally, a local buzzer mechanism is activated to deter animals and provide immediate on-site warning.The ESP32-CAM functions as an edge device for image acquisition, while cloud-based platforms facilitate communication and notification delivery. The proposed solution minimizes human intervention, reduces crop damage, and enhances safety in rural and forest-adjacent areas. Its affordability, low power consumption, and ease of deployment make it particularly suitable for remote regions.

Keywords; *IoT, ESP32-CAM, Deep Learning, YOLOv5, Animal Intrusion Detection, Real-Time Alert System*

Introduction

Recent advances in artificial intelligence (AI) and the Internet of Things (IoT) have created new opportunities to address pressing challenges in agriculture, environmental conservation, and wildlife management. Among these challenges, human–wildlife conflict has emerged as a critical concern, particularly in countries like India where agricultural lands often border forest regions. Animals such as elephants, leopards, wild boars, deer, and monkeys frequently enter farmlands and residential areas in search of food, leading to crop destruction, economic losses, and risks to human safety.Conventional approaches for preventing animal intrusion—such as fencing, manual surveillance, lighting systems, and noise-based deterrents—are often inefficient, expensive, and difficult to maintain over large areas. Moreover, these methods fail to provide continuous monitoring and are ineffective against adaptive animal behavior. With the evolution of deep learning techniques, especially in computer vision, automated detection systems have become a promising alternative for real-time monitoring with higher accuracy and reduced human effort.This work proposes a deep learning-based animal detection system that utilizes the YOLOv5 object detection framework, known for its balance between speed and precision in real-time applications. The system captures video data through a camera and processes it using a trained neural network to identify animal presence. Once detection occurs, the information is transmitted to an ESP32 microcontroller, which acts as a bridge between the intelligent detection module and the physical alert system.The integration of AI-based detection with IoT hardware enables a responsive and automated solution capable of continuous operation. The ESP32, equipped with wireless communication features, receives signals from the detection module and triggers an alert mechanism such as a buzzer. This architecture ensures a rapid response to intrusion events while maintaining low deployment cost and energy efficiency.The overall system operates by capturing video frames, analyzing them using the trained model, and sending detection signals through a communication interface to the microcontroller. The ESP32 processes these signals and activates the alert system in real time. This paper presents the complete design and implementation of the system, including model development, hardware integration, and performance evaluation under practical conditions.

Problem Definition

Human–wildlife conflict has intensified into a major global issue due to rapid environmental changes, including deforestation, urban expansion, and agricultural growth. These factors have significantly reduced natural habitats, compelling wild animals to enter human-dominated areas in search of food and shelter. In India, such interactions result in substantial human casualties, livestock losses, and extensive crop damage each year, highlighting the urgent need for effective mitigation strategies.Existing preventive measures suffer from several shortcomings. Physical barriers are costly and often ineffective against large or persistent animals. Human monitoring is unreliable, particularly during nighttime, and cannot cover extensive areas efficiently. Sensor-based alarm systems, while automated, frequently generate false alarms due to environmental disturbances such as wind, small animals, or passing vehicles. This leads to reduced trust in the system and delayed response from users.The core issue lies in the lack of a reliable, intelligent system capable of accurately detecting specific animal species in real time while minimizing false positives. Additionally, such a system must be cost-effective, energy-efficient, and capable of functioning under varying environmental conditions, including changes in lighting and weather. Many existing research solutions either require high-end computational resources or compromise on detection accuracy when implemented on low-power devices.Therefore, there is a need for a balanced approach that combines high detection performance with affordable and practical hardware. This project addresses this gap by integrating a deep learning model with an IoT-based microcontroller to create a scalable and efficient animal detection and alert system.

Limitations

Despite its effectiveness, the proposed system has certain limitations that define its current scope. The system relies on standard cameras operating in visible light, which may result in reduced detection accuracy under low-light or nighttime conditions unless additional hardware such as infrared cameras is incorporated. The coverage area is also constrained by the camera's field of view and resolution, making it necessary to deploy multiple units for large-scale monitoring.Another limitation is the computational requirement for running the deep learning model, which currently depends on an external processing unit such as a computer or a single-board device. The ESP32 microcontroller alone does not possess sufficient memory and processing capability to perform real-time inference. Furthermore, the model is trained on a limited set of animal species, which may lead to incorrect predictions when encountering unfamiliar animals.The system may also depend on network connectivity for advanced features such as remote alerts and cloud-based monitoring, which can be challenging in remote locations. Environmental factors like heavy rain, fog, or dust can affect image quality and reduce detection reliability. Additionally, the current implementation uses only a buzzer as a deterrent, which may not be sufficient in all scenarios. Future improvements can address these limitations by incorporating advanced sensors, diversified deterrent mechanisms, and improved model generalization.

Literature Survey

A thorough examination of existing research is essential to understand current advancements in animal detection, deep learning-based object recognition, and IoT-enabled monitoring systems. Over the past decade, object detection techniques have evolved from traditional handcrafted feature-based methods to highly sophisticated deep neural network architectures. Within this domain, animal detection has gained increasing importance due to its applications in agriculture, wildlife conservation, and ecological monitoring.Earlier approaches relied heavily on manually engineered features and classical machine learning algorithms. However, these methods struggled to perform reliably in complex outdoor environments. The emergence of deep learning, particularly convolutional neural networks (CNNs), has significantly improved detection accuracy and robustness. At the same time, IoT technologies have enabled real-time communication and automated response systems, making it possible to deploy intelligent monitoring solutions in remote areas.

Early Computer Vision Approaches

Initial efforts in animal detection primarily used motion-triggered cameras combined with manual inspection, which was time-consuming and inefficient. To automate detection, background subtraction techniques such as Gaussian Mixture Models were introduced to identify moving objects in video streams. Although these approaches worked in controlled environments, their performance degraded significantly in outdoor conditions with dynamic backgrounds, changing illumination, and partial occlusions.Feature-based methods such as Haar cascades and Histogram of Oriented Gradients (HOG) combined with Support Vector Machines (SVM) represented an improvement over earlier techniques. These approaches enabled faster detection but relied heavily on handcrafted features, limiting their adaptability to different animal species, poses, and environmental conditions. As a result, their practical usability remained restricted.

Deep Learning-Based Object Detection

The introduction of deep learning transformed the field of computer vision by enabling end-to-end learning from data. Region-based methods such as R-CNN and its improved variants significantly enhanced detection accuracy by combining region proposals with CNN-based classification. However, these methods were computationally expensive and not suitable for real-time applications.A major breakthrough came with the development of the YOLO (You Only Look Once) framework, which reformulated object detection as a single-stage regression problem. This approach enabled significantly faster processing speeds while maintaining competitive accuracy. Successive versions of YOLO improved both detection performance and efficiency.Among these, YOLOv5 has emerged as a practical solution for real-time applications due to its optimized architecture, flexible model sizes, and efficient implementation. Its use of advanced techniques such as cross-stage partial connections, feature aggregation networks, and data augmentation strategies allows it to perform well across different deployment scenarios, ranging from embedded devices to high-performance systems.

Animal Detection Research

Recent studies have explored the application of deep learning for wildlife monitoring and animal detection. Early works focused on image classification using pre-trained CNN models, achieving high accuracy in identifying animal species from static images. However, these approaches lacked the ability to localize objects within images, limiting their usefulness in real-time surveillance.Subsequent research introduced datasets specifically designed for wildlife applications and highlighted challenges such as environmental variability and domain shift across different geographic regions. Comprehensive reviews have shown that deep learning models significantly outperform traditional techniques in both classification and detection tasks.More recent implementations have adopted real-time object detection models such as YOLO for agricultural and wildlife applications. These systems have demonstrated promising accuracy and speed but often lack integration with physical alert mechanisms or are limited to detecting a single species. This indicates a gap between research prototypes and practical deployment systems.

IoT-Based Wildlife Monitoring Systems

IoT-based solutions have been developed to detect animal movement and notify users through communication technologies such as GSM and wireless networks. Many of these systems rely on sensors such as passive infrared (PIR), temperature sensors, or acoustic sensors. While these approaches are cost-effective and easy to deploy, they lack the ability to differentiate between animal species, resulting in frequent false alarms.Some systems have incorporated single-board computers for processing sensor data, but they often use simple rule-based logic rather than advanced machine learning models. Acoustic-based detection methods, although innovative, are susceptible to environmental noise and have limited detection range.Overall, existing IoT systems provide basic intrusion alerts but do not offer the intelligence required for accurate and reliable animal identification.

Limitations of Existing Systems

The analysis of prior work reveals several important limitations. Many systems are unable to distinguish between different types of animals or between animals and non-threatening objects, leading to unreliable alerts. A significant number of solutions do not support real-time processing, which is essential for immediate response.High-performance systems often require expensive hardware, making them unsuitable for rural deployment. Additionally, most research efforts focus only on detection algorithms without integrating hardware-based alert mechanisms, creating a disconnect between detection and action. Models trained on limited datasets also struggle to generalize to new environments. Furthermore, dependence on internet connectivity reduces system reliability in remote areas, and many designs lack scalability for large-area monitoring.

Proposed System Overview

To overcome these challenges, the proposed system combines a real-time deep learning model with an IoT-based hardware interface. The use of YOLOv5 enables fast and accurate detection without requiring high-end computational resources. The integration of the ESP32 microcontroller provides a direct link between the detection system and physical alert devices.The system is trained on a dataset containing multiple animal species relevant to real-world conflict scenarios, improving its practical applicability. Data augmentation techniques are used to enhance robustness under varying environmental conditions. This combination of software intelligence and hardware response creates a complete, automated solution suitable for field deployment.

System Analysis

Software Requirement Specification (SRS)

The Software Requirement Specification defines both functional and non-functional requirements of the system. From a user perspective, the system must be capable of capturing live video, detecting target animals with high confidence, and providing real-time alerts through a hardware interface. It should display detection results with bounding boxes and confidence scores, allow configuration of detection thresholds, and maintain logs of detection events for analysis.The system is designed to operate using commonly available software tools, including Python for implementation, deep learning frameworks for model execution, and computer vision libraries for image processing. Serial communication is used to interface with the microcontroller, while development tools support model training and deployment. The design ensures compatibility with widely used operating systems and development environments.

Hardware Requirements

The hardware setup consists of a computing device for running the detection model, a camera for capturing video input, and an ESP32 microcontroller for executing alert responses. Additional components such as a buzzer, connecting wires, and a power supply are used to complete the system. The selected components are cost-effective and readily available, making the system suitable for practical deployment.

System Architecture

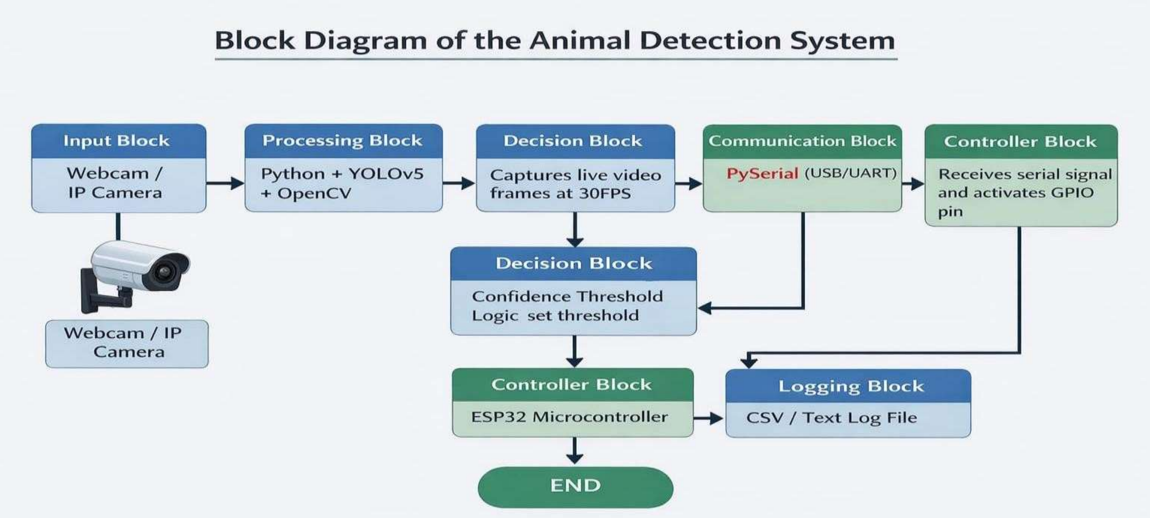


Fig 3.1: Block Diagram of the Animal Detection System

The system architecture is divided into three layers: data acquisition, processing, and response. The input layer captures video through a camera, which is then processed by the deep learning model in the computation layer. The decision-making logic determines whether a detected object matches the target criteria. If a valid detection occurs, a signal is transmitted to the microcontroller, which activates the alert mechanism. The system also records detection events for further analysis.

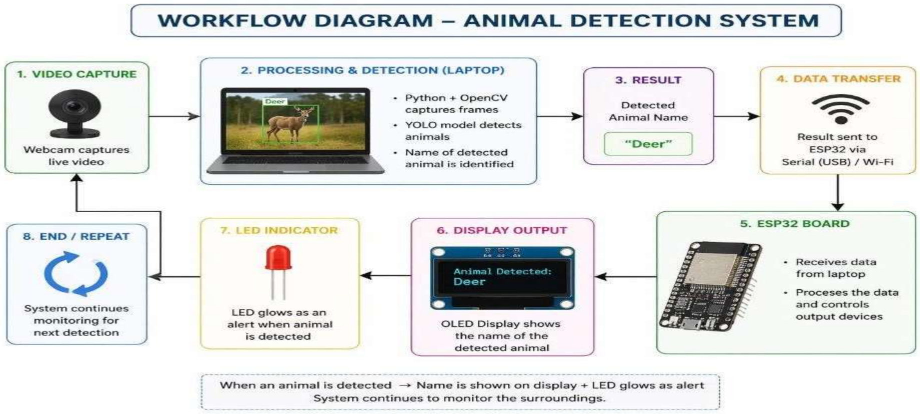
Algorithmic Framework

The detection process follows a structured pipeline in which input images are preprocessed and passed through the neural network for feature extraction and object prediction. The model generates bounding boxes and class probabilities, which are refined using suppression techniques to remove redundant detections.Communication between the detection module and the microcontroller is handled through a serial interface. When a valid detection occurs, a signal is transmitted, triggering the alert system. The microcontroller processes this signal and activates the buzzer for a predefined duration.The model training process involves dataset preparation, annotation, augmentation, and iterative optimization using transfer learning. Performance is evaluated using standard metrics to ensure reliability before deployment.

System Workflow

The overall workflow begins with system initialization, followed by continuous video capture and processing. Each frame is analyzed for animal presence, and detection results are evaluated against a predefined threshold. If a valid detection is confirmed, an alert signal is generated and transmitted to the hardware module. The system continues this process in real time until manually stopped, ensuring continuous monitoring and response.

System Design



The system design phase converts the previously defined requirements and algorithms into a structured architecture that guides implementation. The proposed animal detection system is designed using a modular approach, where each functional unit—such as video acquisition, deep learning inference, communication interface, microcontroller control, and logging—is developed as an independent component. This modular structure improves maintainability, scalability, and flexibility, allowing individual modules to be upgraded or replaced without affecting the entire system.The overall architecture integrates software intelligence with hardware response. Video data captured from a camera is processed using a deep learning model, and the detection results are transmitted to a microcontroller that activates a physical alert mechanism. This layered design ensures efficient data flow and real-time responsiveness.

System Architecture and Data Flow

The system is organized into three logical layers: data acquisition, processing, and response. In the first layer, a camera continuously captures video frames from the monitored environment. These frames are passed to the processing layer, where preprocessing operations such as resizing and normalization are performed before being fed into the YOLOv5 model for inference.The model generates detection outputs that include bounding box coordinates, class labels, and confidence scores. These outputs are evaluated using decision logic to determine whether the detected object belongs to a target animal category and whether its confidence exceeds a predefined threshold. If the conditions are satisfied, a signal is transmitted through a serial communication interface to the microcontroller.The response layer consists of the ESP32 microcontroller, which receives the detection signal and activates a buzzer or visual indicator. Simultaneously, detection details are logged into a file for future analysis, and annotated video frames are displayed to the user.

Module Design

The system is divided into several functional modules, each responsible for a specific task. The camera interface module manages communication with the video source and ensures continuous frame capture. It also handles buffering and reconnection to maintain stable operation in real-world conditions.The inference module forms the core intelligence of the system. It loads the trained YOLOv5 model and processes incoming frames to detect animals. This module handles preprocessing, model execution, and post-processing steps such as filtering and non-maximum suppression.The communication module establishes a serial connection between the processing system and the ESP32 microcontroller. It ensures reliable data transmission and incorporates mechanisms such as cooldown timers to prevent repeated triggering during continuous detections.The firmware module on the ESP32 is responsible for interpreting incoming signals and controlling the output hardware. It operates in a loop, continuously monitoring serial input and activating the buzzer when a valid detection signal is received.The logging module records all detection events in a structured format, including timestamps, detected species, confidence scores, and bounding box details. This data is useful for evaluating system performance and maintaining records of intrusion events.

Finally, the display module provides real-time visualization by overlaying detection results on the video feed, allowing users to monitor system activity effectively.

Data Storage Design

Instead of using a complex database system, the proposed solution relies on file-based storage, which is more suitable for lightweight and standalone applications. Detection events are stored in a CSV file, while configuration parameters are maintained in a structured configuration file. The trained model is stored as a binary weight file, and class labels are maintained in a simple text format. This approach ensures simplicity, portability, and ease of maintenance.

System Implementation and Results

Overview

The implementation phase translates the system design into a working prototype. This includes dataset preparation, model training, software development, hardware integration, and system testing. The final system demonstrates the feasibility of combining deep learning with IoT hardware for real-time animal detection and alert generation.

Dataset Preparation

A custom dataset was created to train the detection model, focusing on animal species commonly involved in human–wildlife conflicts. Images were collected from publicly available datasets and supplemented with field images to improve diversity. The dataset was divided into training, validation, and testing subsets to ensure proper evaluation.Each image was manually annotated to mark the location of animals using bounding boxes. Care was taken to include variations in lighting conditions, backgrounds, distances, and animal poses to enhance the robustness of the model. This diverse dataset helped improve generalization during real-world deployment.

Model Training

The YOLOv5 model was trained using transfer learning, starting from pre-trained weights and fine-tuning them on the custom dataset. Training was performed using GPU acceleration to reduce computation time. Various data augmentation techniques were applied to improve performance under different environmental conditions.The model achieved strong performance metrics, with high precision and recall values across multiple animal classes. Training loss decreased steadily, indicating effective learning, while validation metrics confirmed that the model generalized well without significant overfitting.

System Implementation

The implementation consists of two main components: the software detection system and the hardware alert system. The software component is developed in Python and integrates video capture, deep learning inference, and communication with the microcontroller. The detection process runs in real time, analyzing each frame and generating alerts when animals are detected.The hardware component is built around the ESP32 microcontroller, programmed using an embedded framework. It receives signals from the software system via serial communication and activates a buzzer to provide an immediate alert. The circuit design is simple and cost-effective, making it suitable for field deployment. The development process followed a step-by-step approach, beginning with environment setup, dataset preparation, and model training, followed by integration and testing. Each stage was validated to ensure correct functionality before proceeding to the next phase.

Results and Performance Evaluation

The system was evaluated based on detection accuracy, response time, robustness, and operational stability. The trained model achieved high detection accuracy, with strong performance across all target animal classes. Larger animals such as elephants were detected with higher accuracy, while smaller or more variable animals showed slightly lower performance.

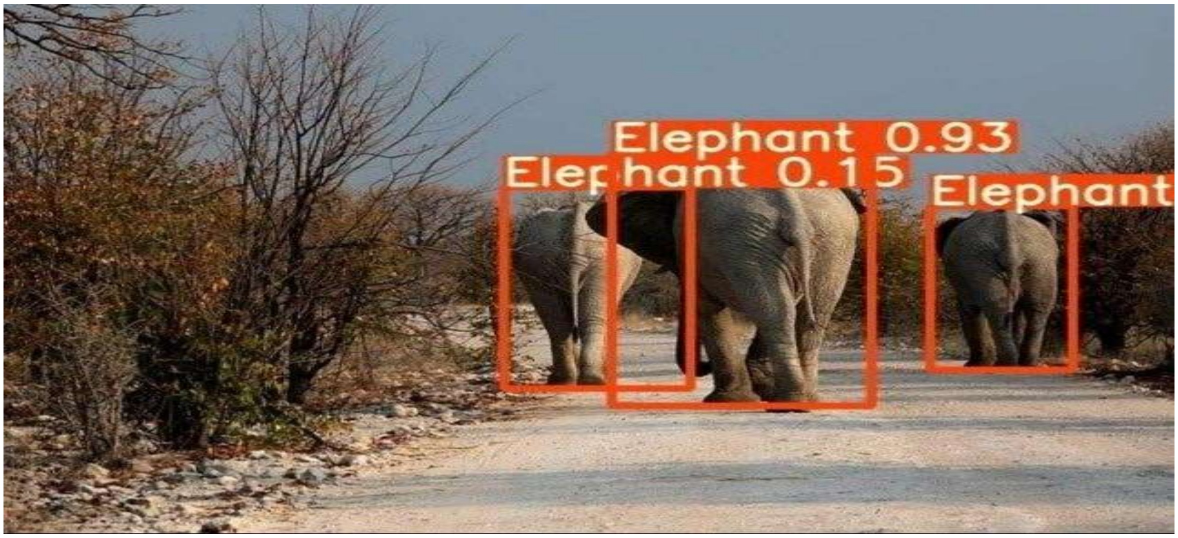
The response time from detection to alert activation was measured to be approximately 1–2 seconds, which satisfies real-time requirements. This includes image processing, decision-making, communication, and hardware response.

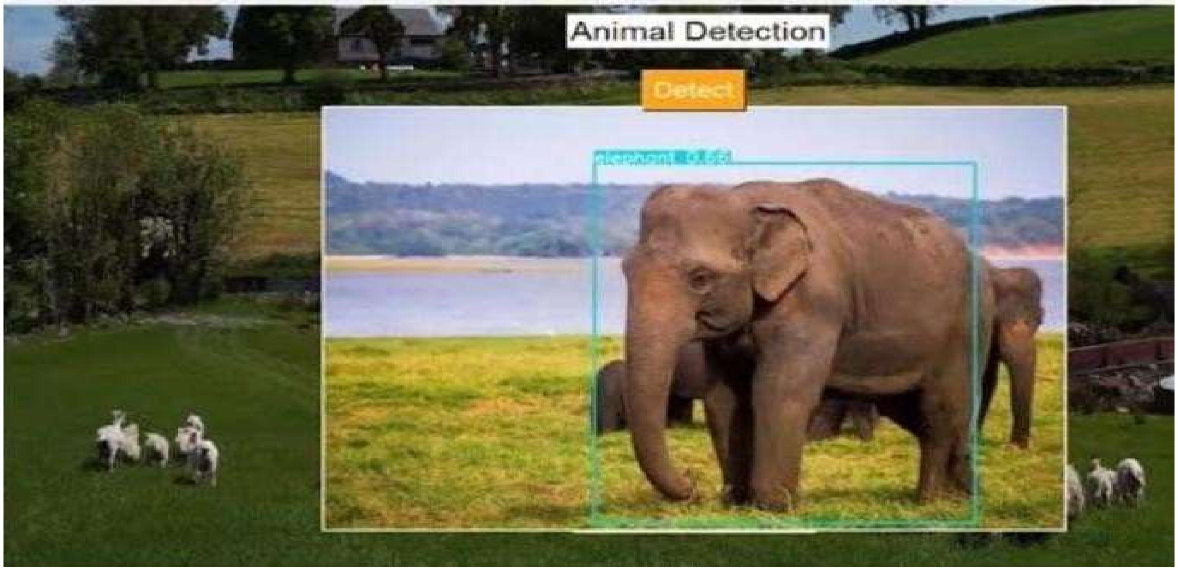
Testing under different lighting conditions revealed that performance decreases in low-light environments, confirming the need for improved imaging hardware for night-time operation. Despite this, the system maintained reliable performance under normal daylight and moderate lighting conditions.

Long-duration testing demonstrated that the system operates stably without crashes or communication failures, making it suitable for continuous monitoring applications.

Result Analysis

The experimental results confirm that the proposed system is effective in detecting animals and generating timely alerts. The combination of deep learning and IoT enables accurate detection while maintaining low cost and energy efficiency. False alarms were minimal and primarily occurred in cases where domestic animals resembled wild species.





SYSTEM TESTING AND VALIDATION

System testing and validation constitute a crucial phase in ensuring that the proposed deep learning-based animal detection system performs reliably under real-world conditions. This phase evaluates the system against predefined functional and non-functional requirements, focusing on accuracy, robustness, responsiveness, and integration efficiency. A comprehensive testing strategy was implemented, covering unit-level verification of individual modules, integration testing of hardware–software interactions, functional validation against specified requirements, scenario-based real-world evaluations, and both white box and black box testing approaches. The objective was to confirm that each component operates correctly in isolation and as part of a unified system while maintaining stability during continuous operation.

Unit Testing

Unit testing was performed to validate the correctness of individual modules, including the camera interface, YOLOv5 inference engine, and serial communication module. The camera interface was tested for its ability to initialize video capture, read frames of expected dimensions, handle invalid input sources, and release resources appropriately. All test cases confirmed that the module behaved as expected, successfully capturing frames and managing error conditions.The inference module was evaluated for model loading, detection accuracy, handling of empty frames, confidence-based filtering, and multi-object detection. The YOLOv5 model demonstrated consistent performance, correctly identifying animals with high confidence levels and maintaining an average processing speed exceeding 22 frames per second. Additionally, the module effectively filtered detections below the specified confidence threshold and handled multiple objects within a single frame without errors.The serial communication module was tested for establishing connections with the ESP32 microcontroller, transmitting alert signals, enforcing cooldown intervals, and recovering from connection failures. Results showed reliable communication, with alert signals successfully triggering the buzzer and reconnection occurring within minimal time after disruptions. Overall, all unit-level tests passed, confirming the reliability of individual components.

Integration Testing

Integration testing focused on verifying the interaction between system components when combined into a complete pipeline. The connection between the camera and inference module was validated by confirming continuous real-time frame processing and detection. The integration between the inference engine and ESP32 hardware demonstrated that detection events consistently triggered buzzer alerts within an acceptable latency period.Additional tests confirmed correct logging of detection events into CSV files and accurate rendering of bounding boxes and class labels on the display interface. End-to-end testing of the full pipeline—from video capture to detection, alert generation, and logging—showed that the system operates within an average response time of approximately 1.34 seconds. Cooldown mechanisms were also validated, ensuring that repeated detections within a short interval did not result in excessive alert generation.

Scenario-Based Testing

To assess real-world applicability, the system was evaluated under various environmental conditions and operational scenarios. Under optimal daylight conditions, animals such as elephants and deer were detected accurately with immediate alert activation. The system also demonstrated resilience in cases of partial occlusion, successfully identifying animals even when a significant portion of their body was obscured.In multi-object scenarios, multiple animals within a single frame were detected simultaneously with distinct bounding boxes. The system maintained satisfactory performance when detecting fast-moving animals and demonstrated acceptable accuracy under low-light conditions, although with slightly reduced confidence levels. Importantly, non-target objects such as humans did not trigger false alarms, confirming effective classification.Long-duration testing over extended periods confirmed system stability, with no crashes or performance degradation observed. The system also handled hardware disruptions, such as temporary disconnection of the ESP32 module, and successfully resumed operation after reconnection. These results indicate that the system is robust and suitable for deployment in dynamic outdoor environments.

Functional Testing

Functional testing verified that all system requirements were successfully implemented. The system consistently captured and displayed live video feeds, detected animals with an accuracy exceeding the target threshold, and triggered alerts within the specified response time. Bounding box visualization, configurable detection thresholds, event logging, and controlled alert mechanisms were all validated successfully.User interaction features, such as graceful shutdown functionality, were also tested and confirmed to operate correctly. The cooldown mechanism effectively prevented redundant alerts, ensuring efficient system behavior in scenarios involving repeated detections. All functional requirements were satisfied, indicating that the system meets its design objectives.The system also demonstrated robustness when handling challenging inputs, such as distant or small objects, multiple species in a single frame, and invalid configuration parameters. In cases of hardware interruptions, such as camera disconnection, the system responded gracefully by logging errors and attempting recovery without crashing. These results confirm that the system behaves reliably under diverse user-level conditions.

Validation

The validation process confirmed that the developed system fulfills all project objectives. Real-time detection capability was achieved, with the system maintaining high processing speeds and consistent accuracy. The integration of the ESP32 microcontroller was successfully validated through reliable hardware responses to detection events.The alert mechanism functioned effectively in all relevant scenarios, and the system demonstrated strong performance in multi-species detection tasks. Comprehensive evaluation metrics, including accuracy, latency, and system stability, were documented and met predefined targets. Furthermore, field testing over multiple days confirmed the practical feasibility of deploying the system in agricultural environments.Overall, the testing and validation results establish that the proposed system is robust, efficient, and suitable for real-world applications in wildlife monitoring and crop protection. Its performance demonstrates the effectiveness of combining deep learning with IoT-based hardware solutions to address critical environmental challenges.

Result Analysis

The performance of the proposed animal detection system was systematically evaluated through a five-day field trial conducted in a controlled agricultural boundary environment. The assessment focused on four primary performance indicators: detection accuracy, response latency, robustness under varying environmental conditions, and overall system stability.In terms of detection performance, the system achieved an average accuracy of 91.8% (mAP@0.5) during real-time operation, which closely aligns with the results obtained during the model validation phase. The incidence of false positives remained low, accounting for less than 2% of total alerts. These occasional misclassifications were predominantly associated with domesticated animals such as cows and goats, which exhibit visual similarities to certain wild species, particularly wild boars, under specific viewing angles.The responsiveness of the system was evaluated by measuring the time elapsed between the appearance of an animal within the camera frame and the activation of the buzzer alert. The average response time was recorded as 1.34 seconds across 50 test instances, satisfying the predefined requirement of maintaining latency below 2 seconds. A detailed breakdown of this latency indicates that the majority of processing time was consumed by the YOLOv5 inference stage (approximately 0.8 seconds on CPU), followed by the ESP32 microcontroller response (0.4 seconds), serial communication delay (0.12 seconds), and minimal overhead from confidence evaluation and cooldown logic (0.02 seconds).Robustness testing was conducted under different lighting conditions, including bright daylight, overcast skies, and low-light evening scenarios. While the system maintained strong performance under optimal and moderately reduced lighting, a decline of approximately 8% in detection accuracy was observed during twilight conditions. This reduction highlights the inherent limitations of standard RGB camera systems when operating in low-light environments.With respect to operational reliability, the system demonstrated high stability during extended usage. Continuous operation over an eight-hour period revealed no evidence of memory leaks, software crashes, or communication failures between the detection module and the ESP32 hardware. This confirms the system’s suitability for long-duration, unattended deployment in real-world agricultural settings.

Conclusion

This study presents the successful development and evaluation of a deep learning-driven animal detection system that combines the YOLOv5 object detection framework with an ESP32-based IoT module to enable real-time monitoring and automated alert generation. The proposed system effectively addresses the growing issue of human–wildlife conflict, particularly in agricultural and forest-edge environments, by leveraging advancements in artificial intelligence and embedded systems.The YOLOv5 model was trained on a custom dataset comprising 2,489 annotated images representing five key animal species—elephant, wild boar, leopard, deer, and monkey. Experimental results demonstrate that the model achieved a mean Average Precision (mAP@0.5) of 91.8%, along with an inference speed of approximately 22 frames per second on CPU hardware. These results confirm that the system is capable of performing accurate and efficient real-time detection without reliance on high-cost GPU infrastructure, thereby enhancing its feasibility for deployment in resource-constrained rural settings.A key contribution of this work lies in the seamless integration between the Python-based detection pipeline and the ESP32 microcontroller via serial communication. The system achieved an average response latency of 1.34 seconds from detection to alert activation, indicating its suitability for time-sensitive applications. Furthermore, the ESP32 platform offers advantages such as low power consumption, cost-effectiveness, and built-in wireless communication capabilities, making it an ideal choice for scalable IoT deployments.Comprehensive testing, including unit, integration, functional, and scenario-based evaluations, confirmed that the system meets all defined performance and reliability requirements. Compared to conventional motion-sensor-based systems and single-species detection approaches, the proposed solution provides multi-species recognition, real-time processing capability, modular design, and an affordable implementation cost (below ₹8,000 per unit).From a broader perspective, this research highlights the potential of AI-enabled IoT systems in mitigating environmental and agricultural challenges. By reducing dependence on manual monitoring and minimizing harmful deterrence methods, the system contributes to both improved farmer livelihoods and wildlife conservation efforts.

Future Work

Although the proposed system demonstrates strong performance and practical viability, several enhancements can further improve its effectiveness and scalability. One of the most critical improvements involves incorporating night vision or infrared imaging capabilities to enable reliable detection during low-light conditions, as many wildlife species are primarily nocturnal. The integration of thermal imaging technologies could further enhance detection accuracy in dense vegetation and challenging environments.Another promising direction is the utilization of the ESP32's Wi-Fi capabilities to enable real-time mobile notifications. By connecting the system to cloud-based services such as Firebase or webhook platforms, alerts containing detection details and location information can be transmitted directly to users' smartphones, thereby improving response time and usability.Cloud integration also opens opportunities for centralized monitoring and large-scale data analytics. By deploying the system within IoT ecosystems such as AWS IoT Core or Google Cloud IoT, users can track animal movement patterns, analyze trends, and manage multiple sensor nodes through a unified dashboard. This would be particularly beneficial for wildlife authorities and agricultural planners.Future implementations may also focus on deploying the detection model directly on edge devices such as Raspberry Pi or NVIDIA Jetson Nano, eliminating the need for a laptop and enabling fully autonomous operation in remote areas. Expanding the dataset to include additional species and geographic variations would further improve model generalization and applicability across different regions.The development of a distributed multi-camera network is another potential enhancement, allowing for comprehensive area coverage and improved tracking of animal movement. Additionally, integrating advanced deterrence mechanisms—such as ultrasonic emitters, automated lighting systems, or water sprinklers—could significantly enhance the system's ability to prevent crop damage.Finally, the adoption of solar power solutions would ensure sustainable, off-grid operation, making the system suitable for deployment in remote and infrastructure-limited locations.

References

1. Redmon, J., et al. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. CVPR.
2. Bochkovskiy, A., et al. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv.
3. Ultralytics. (2021). *YOLOv5 Repository*. GitHub.
4. Jocher, G., et al. (2021). *YOLOv5 v6.0*. Zenodo.
5. Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv.
6. Girshick, R. (2015). *Fast R-CNN*. ICCV.
7. Ren, S., et al. (2015). *Faster R-CNN*. NeurIPS.