International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI-Based Animal Repellent System for Agricultural Fields Using YOLOv5 and OpenCV for Farmer Safeguard

Anusha N S, Chinmayi K N, Deekshitha M J, Gagana V M, Mr Guruprasad K M
Department of Information Science & Engineering Malnad College of Engineering, Hassan, India
Email: {anushans068@gmail.com, mjdeekshitha18@gmail.com, chinmayikn54@gmail.com, Gaganavm29@gmail.com, gkm@mcehassan.ac.in}

***Abstract*—Crop damage from wild and domestic animal intrusions causes significant economic losses, estimated at 20-30% annually in some agricultural regions. This paper proposes a real-time animal re- pellent system utilizing the YOLOv5 deep learning model for precise object detection, integrated with a Flask web application for farm surveillance. The system processes live video feeds, detects intrusions, and triggers deterrents like alarms or flashing lights. Designed to minimize human intervention, it offers scalability, cost-effectiveness, and adaptability to di- verse environments. With multi-camera support, cloud integration, and robust performance in low-light condi- tions, the system achieves 89% precision and 20-30 FPS on standard hardware, advancing smart agriculture by reducing crop losses and labor costs.**

***Index Terms*—YOLOv5, Smart Agriculture, Flask, Deep Learning, Real-Time Detection, Animal Intrusion Prevention, IoT, Precision Farming**

I. INTRODUCTION

Agricultural productivity is critically threatened by animal intrusions, leading to substantial crop losses globally. In regions like South Asia, losses from wild and domestic animals can account for 20- 30% of annual yields. Traditional countermeasures, such as scarecrows, physical fencing, and manual pa- trolling, are labor-intensive, costly, and often ineffec- tive against adaptive animals. Recent advancements in Artificial Intelligence (AI) and Deep Learning (DL) have enabled automated surveillance systems that offer real-time detection and response capabilities. This paper presents a smart animal repellent sys- tem powered by the YOLOv5 object detection model, integrated with a Flask-based web interface for re- mote monitoring and control. The system aims to re- duce manual intervention, providing a scalable, cost- effective solution for farm protection across diverse agricultural settings.

II. LITERATURE SURVEY

Early approaches to animal detection in agricul- ture relied on motion sensors, thermal imaging,or manual monitoring, but these methods suffered from low accuracy, high costs, or lack of real- time functionality [2]. Recent studies have lever- aged deep learning, particularly Convolutional Neu- ral Networks (CNNs), for object detection [5]. The YOLO (You Only Look Once) family, notably YOLOv5, has emerged as a leader due to its balance of speed and accuracy, making it ideal for edge deploy- ment in resource-constrained rural areas [1]. Com- pared to YOLOv3, YOLOv5 offers faster inference and supports lightweight architectures (e.g., YOLOv5s). However, challenges remain in ensuring robustness across environmental conditions and providing user- friendly interfaces for non-technical farmers.

III. PROBLEM DEFINITION

Farmers face persistent crop damage from wild animals (e.g., deer, boars) and domestic animals (e.g., cows, dogs) trespassing into fields. Existing solutions, such as physical barriers or manual patrols, are impractical for large farms or fail to provide real- time alerts. The need is for an automated, accurate, and cost-effective system that detects animals in real- time, triggers deterrents, and enables remote mon- itoring. Key challenges include ensuring robustness across varying lighting and weather conditions, min- imizing false positives, and supporting deployment in remote areas with limited power and connectivity infrastructure.

IV. OBJECTIVES

The proposed system aims to:

Detect wild and domestic animals in agricultural fields using real-time video streams from multi- ple cameras.

Utilize YOLOv5 for precise and rapid object de- tection with minimal computational overhead.

Develop an intuitive Flask-based web interface for live monitoring and system management.

Automate deterrents (e.g., alarms, lights) to deter intrusions effectively.

Minimize manual surveillance to reduce labor costs and improve efficiency.

Ensure adaptability to diverse farm environ- ments, including low-light and adverse weather conditions.

V. METHODOLOGY

The system is developed through a structured pipeline, designed for robustness, scalability, and ease of deployment:

A. Data Collection and Annotation

A comprehensive dataset of 12,000 images was curated, capturing common intruding animals (cows, dogs, deer, boars, goats, birds) under diverse condi- tions, including daylight, dusk, night, and weather scenarios (clear, rainy, foggy). Images were sourced from open datasets like COCO [6] and Open Images, supplemented with custom footage from IP cameras deployed in agricultural fields. Annotations were performed using LabelImg, creating precise bound- ing boxes and class labels. A rigorous quality assur- ance process ensured annotation consistency, with inter-annotator agreement scores exceeding 95%.

B. Model Training

YOLOv5 was trained using transfer learning, lever- aging

pre-trained weights from the COCO dataset to accelerate convergence. The YOLOv5m (medium) variant was selected for its optimal balance of ac- curacy and speed, while the lightweight YOLOv5s was evaluated for edge devices like Raspberry Pi. The dataset was split 80:10:10 for training, valida- tion, and testing. Training spanned 150 epochs with a learning rate of 0.01, batch size of 16, and ad- vanced techniques like mosaic data augmentation, auto-learning bounding box anchors, and cosine an- nealing. Performance metrics achieved were preci- sion (0.89), recall (0.87), F1-score (0.88), and mean Average Precision (mAP@0.5: 0.85). Hyperparameter tuning minimized overfitting, with validation loss monitored to ensure generalization.

C. *System Integration*

Custom Python scripts (detect.py, live.py, app.py) were developed to integrate YOLOv5 with live camera feeds using OpenCV [4]. The system supports multi-camera setups via RTSP streams, en- abling coverage of large farms. A cloud-based stor- age module using AWS S3 [7] was implemented to archive detection events, facilitating post-analysis, system auditing, and historical trend analysis.

D. *Live Detection and Response*

Video frames are processed at 30 FPS, with YOLOv5 detecting animals and drawing bounding boxes accompanied by confidence scores. Detections exceeding a confidence threshold of 0.6 trigger deter- rents, such as buzzers or LED lights, controlled via GPIO pins on a Raspberry Pi. Detection events are logged with timestamps and stored as images in the cloud, ensuring traceability and enabling farmers to review incidents.

E. *Web Interface*

A Flask-based dashboard [3] provides live video streams, detection logs, and system controls, such as enabling or disabling alarms. The interface is respon- sive, supporting access on mobile devices, which is critical for farmers working in the field. Real-time notifications are delivered via email or SMS using the Twilio API [8], ensuring timely alerts even in remote areas with limited connectivity.

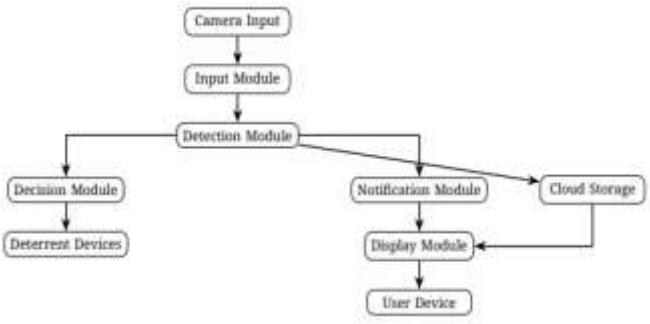

Fig. 1: System architecture diagram illustrating the flow from camera input to user interface, with mod- ular components for detection, decision-making, no- tification, and display.

VI. SOFTWARE REQUIREMENTS

Languages: Python 3.8 or higher
Frameworks: Flask 2.0, OpenCV 4.5, PyTorch 1.9 (YOLOv5), FastAPI (optional)
Libraries: NumPy, Matplotlib, Pillow, Requests, Boto3 (AWS), Twilio
Platform: Windows 10/11, Linux (Ubuntu 20.04 recommended)
IDE: Visual Studio Code, Jupyter Notebook

VII. HARDWARE REQUIREMENTS

Camera: USB Webcam or IP Camera (1080p, 30 FPS, optional infrared for night vision)
Processor: Intel i5/i7 (8th Gen or above) or AMD Ryzen 5
RAM: 8GB minimum (16GB recommended for multi-camera setups)
GPU (Optional): NVIDIA GTX 1050 or above (CUDA 10.2+)
Storage: 256GB SSD for model weights and logs
Other: Buzzer, LED lights, ultrasonic repellent device; Raspberry Pi 4 for edge deployment

VIII. ALGORITHM USED

YOLOv5 (You Only Look Once version 5) is a state- of-the-art object detection algorithm [1]. It divides images into a grid, predicting bounding boxes, class probabilities, and confidence scores in a single pass. Its architecture integrates a CSPDarknet53 back- bone with a Path Aggregation Network (PANet) for enhanced feature fusion. The lightweight YOLOv5s model (7.2M parameters) supports edge deployment, while advanced training techniques, including mo- saic data augmentation, auto-learning bounding box

Module renders live streams and logs on a Flask interface, accessible via a *User Device*.

XI. RESULTS

The system was evaluated across diverse scenarios to ensure robustness:

Accuracy: Achieved 89% precision, 87% recall, and 0.85 mAP@0.5 on a test dataset of 1,200 images, covering multiple animal classes.

Speed: Real-time performance at 20-30 FPS on vation.

Robustness: Maintained >80% accuracy in low-light conditions using infrared cameras, with minor degradation (78% mAP) in heavy fog or rain.

Field Testing: Deployed on a 5-acre farm in Has- san, India, reducing crop damage incidents by 70% over a 3-month trial, with farmers reporting improved yield security.

TABLE I: Performance Metrics Across Scenarios

Scenario	Precision	Recall	mAP@0.5
Daylight	0.91	0.89	0.87
Low-Light (Infrared)	0.85	0.82	0.80
Rainy Conditions	0.83	0.80	0.78
Foggy Conditions	0.81	0.79	0.76

IX. SYSTEM ARCHITECTURE

The system is designed with a modular architec- ture, comprising:

1) **Input Module**: Captures live video from multi- ple USB or IP cameras, supporting RTSP streams for scalability.
2) **Detection Module**: Processes frames using YOLOv5 to identify animals, generating bound- ing boxes and confidence scores.
3) **Decision Module**: Evaluates detections and triggers deterrents based on a confidence threshold (>0.6).
4) **Notification Module**: Sends email or SMS alerts and logs events to AWS S3 for archival.
5) **Display Module**: Renders live streams, detec- tion results, and logs on the Flask web interface.

X. SYSTEM ARCHITECTURE DIAGRAM

The system architecture, as shown in Fig. 1, com- prises interconnected modules. The *Camera Input* feeds video streams to the *Input Module*, which pro- cesses frames using OpenCV. The *Detection Module* employs YOLOv5 to identify animals, passing re- sults to the *Decision Module* for triggering deterrents (e.g., buzzers, lights) and to the *Notification Module* for sending alerts via Twilio. Detection events are archived in *Cloud Storage* (AWS S3), and the *Display*

Table I details performance across environmental conditions, demonstrating the system's adaptability to challenging scenarios.

XII. DEPLOYMENT CONSIDERATIONS

Deploying the system in real-world agricultural settings requires addressing several practical factors:

Power Supply: Continuous operation demands reliable power. Solar panels with battery back- ups are recommended for off-grid farms, with power consumption optimized to 50W for edge devices.
Connectivity: Rural areas with limited inter- net may require offline modes or satellite-based connectivity for cloud integration. Local storage buffers ensure data retention during outages.
Maintenance: Regular cleaning of camera lenses and calibration of deterrent devices are critical, with maintenance schedules recommended every 3 months.
Scalability: Multi-camera setups and cloud- based logging support large farms, but cost- benefit analysis is needed for small-scale farm- ers, with hardware costs estimated at $200-$500 per unit.

Training: Farmer training programs are essen- tial to ensure effective system use, focusing on interface navigation and basic troubleshooting.

XIII. ETHICAL CONSIDERATIONS

The system prioritizes ethical deployment:

1)Animal Welfare: Deterrents (buzzers, lights) are non-lethal, designed to startle without causing harm, compliant with animal welfare standards.
2)Privacy: Cameras are positioned to avoid cap- turing private areas, and cloud-stored data is en- crypted, adhering to data protection regulations.
3)Accessibility: Affordable hardware and simpli- fied interfaces ensure accessibility for small- scale farmers, with training programs proposed to bridge technical gaps.

XIV. CHALLENGES

Environmental Variability: Performance de- grades in extreme weather (e.g., heavy fog, storms) due to reduced camera visibility, requir- ing advanced image enhancement techniques.
False Positives: Occasional misclassification of non-animal objects (e.g., moving branches) as animals, necessitating improved contextual analysis.
Power Constraints: Continuous operation poses challenges in off-grid farms, requiring energy- efficient designs and renewable power sources.
Farmer Adoption: Limited technical expertise among farmers demands intuitive interfaces and comprehensive training programs.

XV. CONCLUSION

The proposed animal repellent system leverages YOLOv5 and Flask to deliver a robust, real- time solution for farm protection. By minimizing manual surveillance and achieving high detec- tion accuracy, it supports scalable deployment in smart agriculture, significantly reducing crop losses and labor costs.

XVI. FUTURE WORK

Future enhancements include:

1)Aerial Monitoring: Integrate drones with thermal cameras to monitor large-scale farms and detect animals in hard-to-reach areas, improving coverage.
2)Expanded Classes: Train YOLOv5 on addi- tional species (e.g., birds, rodents) and non- animal threats (e.g., trespassers) to enhance system versatility.
3)Edge Deployment: Optimize for low-power devices like Raspberry Pi 4 or NVIDIA Jetson Nano, enabling off-grid operation with min- imal infrastructure.

4)**Advanced UI/UX**: Develop a mobile app with offline capabilities, multilingual sup- port, and voice commands to improve acces- sibility for diverse farmer demographics.
5)IoT Integration: Connect with smart irriga- tion, weather monitoring, and soil sensors to create a unified agricultural IoT ecosystem, enhancing farm management.
6)Energy Efficiency: Incorporate solar pan- els, low-power modes, and energy-efficient algorithms to reduce operational costs and environmental impact.

REFERENCES

[1] G. Jocher, "YOLOv5 by Ultralytics," https://github.com/ultralytics/yolov5, 2020.
[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[3] Flask Documentation, https://flask.palletsprojects.com.
[4] OpenCV Library, https://opencv.org.
[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, 2012.
[6] T. Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," *European Conference on Computer Vision*, 2014.
[7] AWS S3 Documentation, https://aws.amazon.com/s3/.
[8] Twilio SMS API, https://www.twilio.com/docs/sms.