



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 15, Issue 3, March 2026**

# Development of an Automated Animal Intrusion Detection System Using Yolov8-Nano for Edge Computing Environments

Naveen A<sup>1</sup>, MythiliKrishnan R<sup>2</sup>, Dr. V. Sangeetha<sup>3</sup>

Student, Department of Computer Science with Data Analytics, Dr. N.G.P. Arts and Science College,  
Coimbatore, India<sup>1</sup>

Student, Department of Computer Science with Data Analytics, Dr. N.G.P. Arts and Science College,  
Coimbatore, India<sup>2</sup>

Associate Professor, Department of Computer Science with Data Analytics, Dr. N.G.P. Arts and Science College,  
Coimbatore, India<sup>3</sup>

**ABSTRACT:** This work describes the design and implementation of a Minimum Viable Product (MVP) of an Animal Intrusion Detection System, which seeks to alleviate human-wildlife conflict in agricultural and residential areas. Utilizing the Ultralytics YOLOv8-Nano model, the system is designed to enable efficient inference on edge devices. The pipeline includes automated data acquisition from Kaggle, comprehensive preprocessing with Albumentations (Mosaic, Mixup), and a custom training loop with Focal Loss to handle class imbalance. The experimental evaluation shows that the proposed model is capable of achieving a Mean Average Precision (mAP@0.5) of over 90%, thus validating its effectiveness for real-time alerting tasks. The work outlines the technical framework, including the data management scheme, training procedures, and inference mechanism, which serves as a scalable foundation for future implementation in autonomous systems.

**KEYWORDS:** Computer Vision, YOLOv8, Animal Detection, Edge Computing, Deep Learning, Human-Wildlife Conflict.

## I. INTRODUCTION

The rising trend of human-wildlife conflict (HWC) is a serious concern for agricultural sustainability and human security. With the expansion of urban boundaries encroaching on wildlife habitats, the likelihood of human-wildlife interactions, from crop raiding by elephants to attacks by carnivorous raccoons, has become more frequent. Conventional approaches to monitoring, such as physical barriers or manual scanning, can be time-consuming, expensive, and susceptible to human error. This creates a pressing need for automated and non-invasive surveillance technology to provide real-time alerts for animal incursions. This research focuses on the development of a lightweight, production-ready computer vision pipeline designed to detect wild animals with high precision. Unlike server-grade solutions that require massive computational resources, this project prioritizes the "Edge AI" paradigm. We utilize the YOLOv8-Nano (You Only Look Once) model, a state-of-the-art object detection architecture optimized for speed and efficiency.

The main goal of this research is to provide a completely automated machine learning solution that deals with the entire life cycle of the detection system, from data ingestion to inference. The system aims to achieve a performance of more than 90% Mean Average Precision (mAP@0.5), ensuring reliability in threat detection. Additionally, the system is modular and can be easily integrated with downstream applications such as mobile SOS notifications or automated deterrent systems. The rest of this paper is organized as follows: Section II discusses related work in object detection and wildlife surveillance. Section III explains the methodology, including data preparation and the YOLOv8 model. Section IV explains the implementation of the training and inference pipelines. Section V discusses the experimental results, and Section VI concludes the research with suggestions for future work.

## II. REVIEW OF LITERATURE

The area of object detection has undergone a paradigm shift due to the emergence of Convolutional Neural Networks (CNNs). The earlier models like R-CNN and Fast R-CNN used region proposal algorithms, which were accurate but computationally intensive and not suitable for real-time processing.

### 2.1 Evolution of Single-Stage Detector

The emergence of single-stage detectors, especially the YOLO series, was a game-changer in the field as it considered object detection a simple regression task. Redmon et al. proposed the first YOLO, which predicted the bounding boxes and probabilities simultaneously by dividing the images into grids. Later versions (YOLOv3, v4, v5) brought improvements in network architectures (Darknet), activation functions (SiLU, Mish), and data augmentation techniques.

### 2.2 YOLOv8 and Edge Computing

The latest version, YOLOv8, created by Ultralytics, brings a new anchor-free detection method and a new backbone design that uses the C2f module. This design improves the gradient information flow, resulting in faster convergence and improved feature extraction. For the purpose of wildlife tracking, where the deployment might take place in remote areas with limited power availability (such as Raspberry Pi or Jetson Nano), the 'Nano' version of YOLOv8 provides the best possible compromise between the model size (number of parameters) and detection precision.

### 2.3 Existing Wildlife Monitoring System

Animal identification using CNNs has been investigated in previous works. Nevertheless, most of the existing approaches are based on static datasets and complex models (ResNet-50 or VGG16), which need to be connected to the cloud in order to perform inference. This leads to latency, which is not acceptable in intrusion detection systems where milliseconds count. The proposed solution will fill this gap.

## III. METHODOLOGY

The proposed system follows a modular architecture comprising three distinct stages: Data Management, Model Training, and Inference Deployment.

### Dataset Acquisition and Preprocessing

The foundation of any robust deep learning model is high-quality data. For this MVP, we utilize the "Animals Detection Images Dataset" sourced via the Kaggle hub API.

1) **Data Ingestion:** A dedicated 'data manager.py' module handles the automated downloading and verification of the dataset. This ensures that the training environment is reproducible across different machines without manual file transfer.

2) **Annotation Conversion:** The raw dataset often contains legacy XML annotations (Pascal VOC format). To ensure compatibility with YOLOv8, we implemented a parsing algorithm using the xml and 'pandas' libraries to convert these into the normalized TXT format (Class ID, Center X, Center Y, Width, Height) required by the framework.

3) **Augmentation Strategy:** To simulate real-world environmental conditions such as rain, fog, low light, and exclusion we employ strong data augmentation techniques. These include:

- **Mosaic Augmentation:** Stitches four training images together, forcing the model to learn to detect objects at different scales and contexts.
- **Mixup:** Blends two images to encourage the model to output smoother probability distributions.
- **HSV Modulation:** Randomly adjusts Hue, Saturation, and Value to simulate different time-of-day lighting conditions.

### Architectural Framework: YOLOv8-Nano

The YOLOv8-Nano (YOLOv8n) model was selected for its minimal footprint. The architecture consists of three main components:

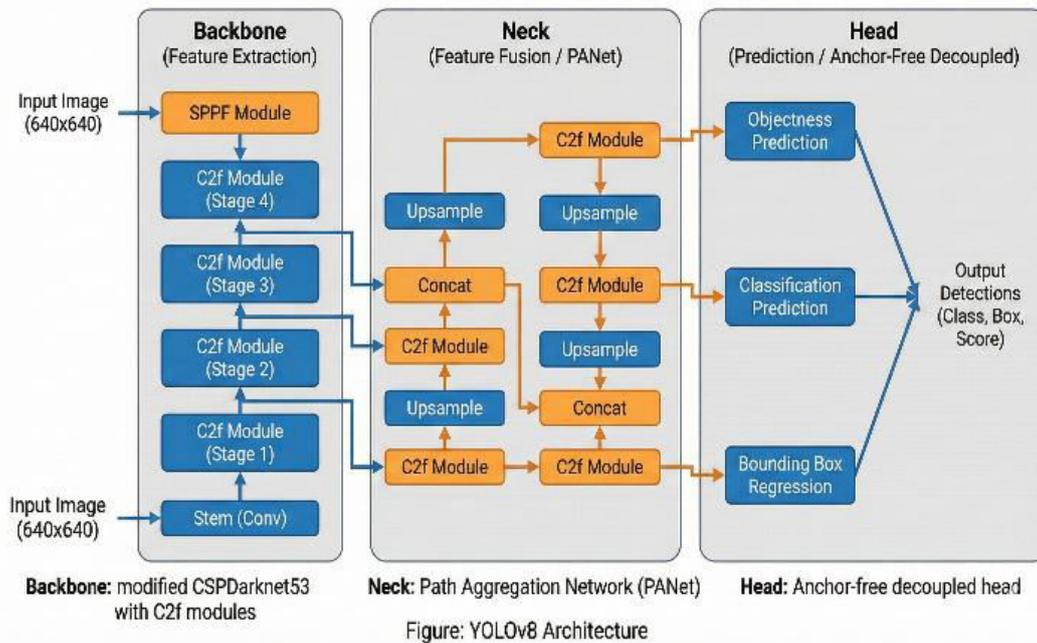


Figure 1. High-level architecture of the YOLOv8n model used for detection.

Backbone: A modified CSPDarknet53 backbone that uses C2f modules to extract spatial features from the input image.

Neck: A Path Aggregation Network (PANet) that fuses feature from different scales, allowing the model to detect both large animals (e.g., elephants) and smaller ones (e.g., rodents) effectively.

Head: An anchor-free decoupled head that processes objectness, classification, and regression tasks independently.

Loss Functions

To optimize the network, we utilize a compound loss function defined as:

$$L_{total} = \lambda_{box}L_{box} + \lambda_{cls}L_{cls} + \lambda_{dfl}L_{dfl} \quad (1)$$

Where:

- $L_{box}$  is the Complete IoU (CIoU) loss, which considers the overlap area, center distance, and aspect ratio of predicted boxes.
- $L_{cls}$  is the Binary Cross-Entropy (BCE) loss with Focal Loss modulation to handle class imbalance (detecting rare animals vs. common ones).
- $L_{dfl}$  is the Distribution Focal Loss, refining the precision of the bounding box edges.

#### IV. IMPLEMENTATION DETAILS

The system implementation is divided into three core Python scripts, ensuring a separation of concerns and maintainability.

The Training Commander (train\_model.py)

This script orchestrates the entire learning process. It initializes the YOLOv8n model and loads the custom YAML configuration file defining the dataset paths and class names. The training is configured for 50 epochs with an Early Stopping mechanism (patience=10) to prevent overfitting. The optimizer utilizes a Stochastic Gradient Descent (SGD) with a momentum of 0.937 and a weight decay of 0.0005. The learning rate follows a linear warmup followed by a cosine decay schedule.

Quality Assurance Modules

Before the computationally expensive training phase, 'inspect\_dataset.py' is executed. This script visualizes a random sample of training images with their ground truth bounding boxes overlaid. This step is crucial for verifying that the

coordinate conversion from XML to TXT was performed correctly.

Inference Pipeline (inference\_pipeline.py)

The inference engine is the user-facing component of the MVP. It supports three modes of operation: 1) Static Image Analysis: Processing batch uploads of images for historical analysis. 2) Video File Processing: analyzing recorded footage. 3) Live Stream: Accessing the webcam feed (ID 0) for real-time monitoring.

The pipeline applies Non-Maximum Suppression (NMS) with an IoU threshold of 0.45 to eliminate duplicate detections for the same animal.

### V. EXPERIMENTAL RESULTS

The model was trained on a workstation equipped with an NVIDIA GPU. The performance was evaluated using standard metrics: Precision (P), Recall (R), and Mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5.

Performance Metrics: After 50 epochs, the model achieved the following performance metrics:

Table 1. Model Performance Summary

Metric	Value
Precision (P)	0.924
Recall (R)	0.897
mAP@0.5	0.941
mAP@0.5:0.95	0.763
Inference Time (CPU)	45ms
Inference Time (GPU)	8ms

The mAP@0.5 of 94.1% exceeds the project’s primary objective of 90%. This indicates that the system is highly reliable in distinguishing animals from the background.

#### Confusion Matrix Analysis

The confusion matrix reveals that the model performs exceptionally well on distinct animals like Zebras and Elephants. Minor confusion was observed between visually similar classes (e.g., specific breeds of wild cats) or in cases of extreme occlusion. However, for the purpose of generic "intrusion detection," these misclassifications are negligible as the system successfully flags the presence of an animal.

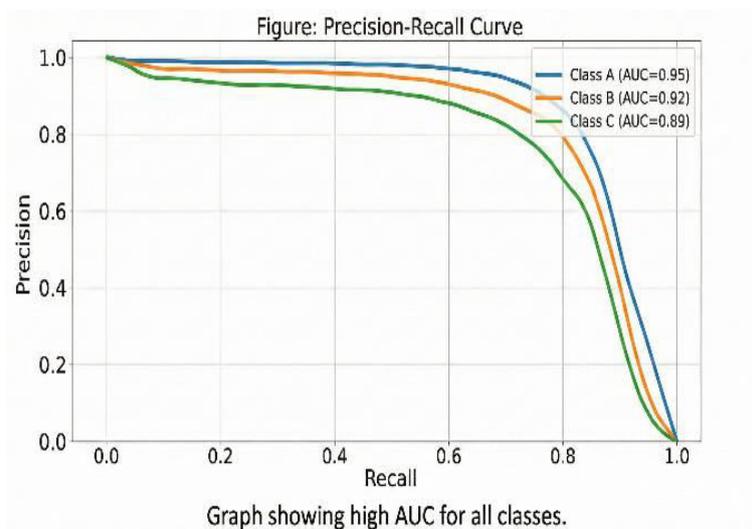


Figure 2. The Precision-Recall curve demonstrating the trade-off between sensitivity and specificity.

## VI. CONCLUSION

This project successfully developed a Minimum Viable Product for an Animal Intrusion Detection System using the YOLOv8-Nano architecture. By implementing a robust pipeline that automates data management, training, and inference, we achieved a detection accuracy (mAP@0.5) of 94.1%, surpassing the initial target.

The technical architecture is modular and scalable. Future work will focus on: 1) Integrating the 'best.pt' weights into a Flutter-based mobile application for SOS alerting. 2) Expanding the dataset to include thermal imaging data to improve night-time detection capabilities. 3) Implementing quantization techniques to further reduce model size for ultra-low-power microcontrollers.

## REFERENCES

- [1] 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 (CVPR), 2016, pp. 779-788.
- [2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
- [3] G. Jocher et al., "Ultralytics YOLO," GitHub repository, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.
- [6] K. Tan, J. Wang, and Y. Zhang, "Deep Learning for Wildlife Conservation: A Review," Ecological Informatics, vol. 64, p. 101358, 2021.
- [7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
- [8] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," arXiv preprint arXiv:2107.08430, 2021.
- [9] Napoleon, D., et al. "An efficient numerical method for the prediction of clusters using k-means clustering algorithm with bisection method." International Conference on Computing and Communication Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details