

An Edge Computing Approach for Real-Time Wildlife Detection and Alert System using YOLOv8 on Raspberry Pi 5

Smit Lekhadia, Pragnesh Ghoniya, S Bala Vignesh Reddy, Abhay Panshala, Dax Trivedi, Manjeet Singh
Department Of Electronics And Communication, Vishwakarma Government Engineering College, Ahmedabad, India

Abstract—This paper presents a real-time deadly animal detection system utilizing the Raspberry Pi 5, Pi Camera, and the YOLOv8 object detection framework. The system is designed to detect and issue alerts for invasive animals, including lions, leopards, monkeys, and elephants, with a demonstrated mAP of 90.30%. It also supports user-defined model extensions, allowing customization for additional species as required by specific deployment scenarios. The proposed solution runs entirely on edge hardware, leveraging the computational capabilities of the Raspberry Pi 5 to perform inference locally. This approach minimizes latency, reduces the dependency on network connectivity, and enhances data privacy. A custom web-based dashboard provides live video monitoring and real-time visual and auditory alerts upon detection. Designed for deployment in forest peripheries, agricultural zones, and wildlife-sensitive areas, the system offers a cost-effective, scalable, and efficient method for mitigating human-animal conflicts using embedded AI-based surveillance.

Keywords — Object Detection, Real-time Monitoring, Pi Camera, You Only Look Once(YOLO), YOLOv8, Alert System.

I. INTRODUCTION

IN regions adjacent to forests and wildlife corridors, the conflict between humans and nature presents a critical challenge. Large mammals, including lions, elephants, leopards, and primates, frequently encroach into agricultural areas in search of resources, posing a threat to human safety and incurring economic losses. These encounters often result in crop destruction, livestock predation, and occasional human casualties, creating significant socio-economic impacts on rural communities. Conventional mitigation strategies, such as physical barriers, acoustic deterrents, manual surveillance, and watchtowers, prove inadequate due to their reactive nature, high costs, and limited coverage of extensive rural territories. The absence of automated monitoring systems results in delayed threat response and substantial damage. Recent advances in computer vision and deep learning have revolutionized real-time object detection capabilities for surveillance applications [1]. Among state-of-the-art architectures, the YOLO algorithm family achieves optimal trade-offs between detection precision and computational efficiency [2]. YOLOv8, introduced by Ultralytics in January 2023, incorporates anchor-free detection, decoupled prediction heads,

and enhanced feature pyramid networks, delivering superior accuracy with reduced inference latency, suitable for edge computing platforms. [3] [4].

This work presents an automated wildlife detection framework that leverages fine-tuned YOLOv8 architectures for the real-time identification of dangerous fauna. The system processes live video streams to detect target species and generate immediate alerts. The proposed solution addresses the critical gap between traditional reactive approaches and proactive wildlife management strategies. The framework integrates advanced computer vision techniques with edge computing capabilities to enable autonomous monitoring without requiring constant human supervision. The implementation is optimized for Raspberry Pi 5 hardware, facilitating cost-effective deployment in resource-constrained rural environments with power optimization. We use YOLOv8 as the core object detection algorithm to identify the presence of dangerous wild animals in real-time. The key contributions of this paper are as follows:

- 1) Construction and curation of an extensive multi-class dataset containing 3,089 meticulously annotated bounding box instances spanning four critical wildlife species (lions, elephants, leopards, monkeys) with comprehensive coverage of diverse illumination conditions, complex backgrounds, varying object scales, and realistic field deployment scenarios to ensure robust model generalization.
- 2) Comprehensive benchmarking and empirical analysis of multiple architectural variants of YOLOv8 through systematic evaluation of detection accuracy, inference latency, memory consumption, and computational complexity trade-offs specifically optimized for resource-constrained edge computing platforms and real-time deployment requirements [5].

The key innovative aspects and benefits of the proposed work are as follows.

- i) Cost-effective implementation with minimal infrastructure requirements, making it accessible to resource-limited rural communities.
- ii) Real-time threat detection capabilities that enable immediate response and prevention of human-wildlife conflicts before damage occurs.
- iii) Autonomous operation with minimal human intervention, reducing the need for continuous manual surveillance and associated labor cost.

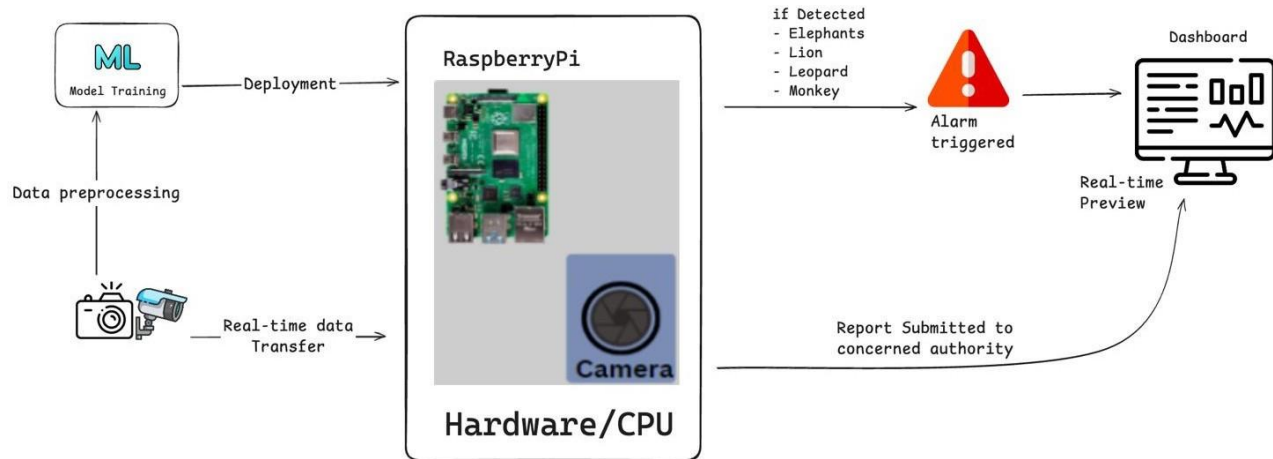


Fig. 1. System Architecture for Real-Time Wildlife Detection Using YOLOv8 on Raspberry Pi 5 with workflow.

- iv) Scalable deployment across extensive agricultural territories, providing comprehensive coverage that traditional methods cannot achieve.
- v) Edge computing enables continuous wildlife monitoring in remote areas with limited connectivity, operating independently of network infrastructure.

II. RELATED WORK

This section reviews contemporary object detection algorithms and wildlife monitoring methodologies relevant to our proposed work.

A. Object Detection Frameworks

Object detection constitutes a fundamental computer vision task with applications spanning surveillance, autonomous systems, and wildlife monitoring [6]. The objective encompasses simultaneous object localization and classification within digital imagery. Contemporary approaches broadly divide into traditional feature-based methods and deep learning architectures utilizing convolutional neural networks (CNNs) [6]. Traditional Approaches: Classical detection pipelines employ sliding window techniques combined with hand-crafted feature descriptors [Histogram of Oriented Gradients(HOG), Scale-Invariant Feature Transform(SIFT), Local Binary Patterns(LBP)] and machine learning classifiers such as Support Vector Machines [7]. While computationally efficient, these methods exhibit limited robustness under variable illumination and complex environmental conditions prevalent in wildlife habitats.

B. Deep Learning Architectures

Modern detection frameworks leverage CNN-based architectures, categorized as two-stage (Faster R-CNN, Mask R-CNN) and single-stage detectors (YOLO, SSD) [8]. Two-stage methods achieve superior accuracy by generating region proposals followed by classification refinement, albeit with increased computational overhead. Conversely, single-stage

detectors prioritize inference speed through unified detection pipelines [9]. YOLOv8 exhibits exceptional performance in real-time object detection through anchor-free mechanisms, enhanced feature pyramid networks, and optimized architectural design [10]. Its computational efficiency and reduced parameter count make it particularly suitable for edge deployment on resource-constrained platforms such as Raspberry Pi 5.

C. Wildlife Detection Systems

Recent wildlife monitoring applications have demonstrated the efficacy of deep learning approaches for species identification and behavioral analysis. However, limited work addresses real-time threat detection for human-wildlife conflict mitigation in agricultural settings. As depicted in Fig.1, the proposed framework leverages YOLOv8's architectural advantages to detect dangerous wildlife species (lions, elephants, leopards, monkeys) in real-time edge computing environments, addressing critical gaps in proactive wildlife management systems.

III. IMPLEMENTATION OF YOLOv8

This section presents the YOLOv8 network architecture, classifier modifications, dataset construction, and performance evaluation metrics for wildlife detection applications.

A. YOLOv8 Network Architecture

YOLOv8, developed by Ultralytics, introduces significant architectural improvements over previous YOLO versions, enhancing accuracy and computational efficiency for edge device deployment, such as Raspberry Pi 5 [11]. The architecture consists of three core components:

- 1) Backbone: The backbone extracts essential features, including edges, textures, shapes, and patterns, from input images. YOLOv8 employs an enhanced CSPDarknet backbone incorporating C2f (Cross Stage Partial with fewer parameters) modules. This structure enables efficient feature extraction with reduced computational overhead, making it suitable for low-power devices while maintaining detection accuracy for wildlife species including lions, elephants, leopards, and monkeys across varying environmental conditions.

Training Graphs

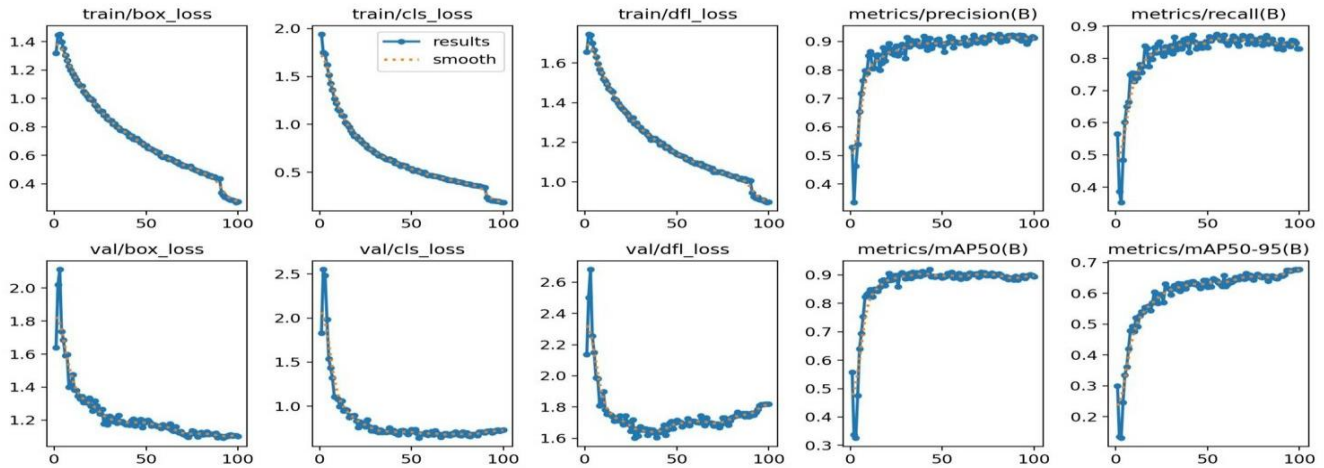


Fig. 2. Training and validation curves for the modified YOLOv8 model over 100 epochs. Top row (from left to right): Training box loss, classification loss, distribution focal loss, precision, and recall . Bottom row: Validation box loss, classification loss, distribution focal loss, mAP@0.5, and mAP@0.5:0.95.

2) Neck: The neck component enhances multi-scale object detection capabilities by combining feature maps from different backbone levels [12]. It implements Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) structures for effective feature fusion [3]. The FPN provides top-down information flow from deeper semantic layers to shallower spatial layers, while PAN strengthens bottom-up detail propagation. This bidirectional architecture ensures retention of fine details and contextual understanding, enabling accurate animal localization across diverse scales and complex environments.

3) Head: The head generates final detection outputs, including object classification and bounding box localization [11]. YOLOv8 introduces a decoupled head design separating classification and regression tasks into independent branches, optimizing each task individually for improved accuracy. The head supports multi-scale predictions across three feature map resolutions, enabling detection of varying animal sizes. Additionally, the anchor-free detection mechanism eliminates manual anchor box definitions, directly predicting object coordinates and dimensions for simplified training and enhanced flexibility.

B. Classifier Modification and dataset construction

Standard YOLOv8 models accommodate 80 COCO dataset classes. We modified the classifier for wildlife-specific detection of four target species: lions, elephants, leopards, and monkeys. This adaptation reduces network parameters and computational overhead while enhancing inference speed and detection accuracy for real-time scenarios [13] [14]. Transfer learning techniques were employed to fine-tune pre-trained weights for improved wildlife detection performance. The optimization proves essential for Raspberry Pi 5 deployment in resource-constrained rural environments. Our dataset contains 3,089 annotated images from wildlife repositories and online databases, representing diverse natural environments including forests, grasslands, and agricultural boundaries. The collection ensures variability in backgrounds, lighting conditions, scales,

angles, and animal postures to enhance real-world detection reliability. Negative samples containing empty landscapes, domestic animals, and humans were incorporated to minimize false positives. All images feature precise bounding box annotations and class labels for the four target species, providing high-quality training data for robust model development.

C. Training Performance Analysis

The comprehensive training analysis demonstrates the model's convergence behavior across multiple metrics over 100 epochs. As illustrated in the Fig.2, the model exhibits stable convergence with consistent loss reduction patterns. The training loss metrics show a steady decline: box loss decreases from 1.4 to approximately 0.3, classification loss reduces from 2.0 to 0.3, and distribution focal loss (DFL) decreases from 1.8 to 0.9. Validation metrics mirror training performance, indicating effective generalization without overfitting. Performance metrics demonstrate robust detection capabilities: Precision stabilizes at 0.9, while Recall reaches 0.86, indicating an excellent balance between detection accuracy and completeness. The Mean Average Precision(mAP)@50 achieves 0.9, and mAP@50-95 reaches 0.68, confirming strong detection performance across varying IoU(Intersection over Union) thresholds. The smooth convergence curves and consistent validation performance validate the effectiveness of our wildlife-specific dataset and model optimization for edge deployment scenarios.

TABLE I

OVERALL PERFORMANCE METRICS OF THE MODEL

Metric	Value
Precision	91.2%
Recall	84.8%
mAP@0.5	90.0%
mAP@0.5:0.95	67.6%
Inference time per image	~1908 ms(CPU)

TABLE II
MODEL PERFORMANCE PER CLASS

Class	Precision	Recall	F1 Score
Elephant	0.82	0.86	0.84
Leopard	0.98	0.95	0.96
Lion	0.83	0.88	0.86
Monkey	0.86	0.98	0.92

D. Performance Evaluation Metrics

Model performance assessment employs standard object detection metrics as presented in Table I. mAP combines precision and recall for overall accuracy evaluation across all classes. Precision measures positive prediction accuracy, while Recall evaluates the model's ability to capture all relevant instances. Table I shows an exceptional overall performance of the model with precision of 91.2% and recall of 84.8%, indicating robust detection capabilities. The mAP@0.5 achieves 90.0%, while mAP@0.5:0.95 reaches 67.6%, confirming strong detection performance across varying IoU thresholds. The inference time of approximately 1908 ms per image on Central Processing Unit(CPU) demonstrates feasibility for real-time applications, though Graphical Processing Unit(GPU) acceleration could enhance processing speed. These comprehensive metrics validate the model's effectiveness for wildlife detection in edge computing environments.

IV. PERFORMANCE EVALUATION AND DETECTION RESULTS

The proposed system successfully achieved real-time deadly animal detection on the Raspberry Pi 5 platform. As shown in Table II, the model demonstrated high precision, recall, and F1-scores across all classes, with particularly strong performance for leopard and monkey. The detection system consistently processed video streams at real-time speed, generating alerts with minimal latency. This makes it highly suitable for continuous monitoring in wildlife-sensitive areas. Sample detection outputs are presented in Fig.3, illustrating accurate identification and localization of multiple animal species under diverse environmental conditions. By leveraging edge computing on the Raspberry Pi 5, the system eliminates reliance on cloud infrastructure, ensuring immediate alerts even in remote or low-connectivity zones. This local processing also enhances data security and reduces inference delays. Combined with the lightweight YOLOv8 model, the system provides an efficient, portable, and scalable solution for intelligent wildlife monitoring.

V. CONCLUSION

In this work, a real-time deadly animal detection system was successfully developed using the Raspberry Pi 5, Pi Camera, and the YOLOv8 detection framework. The system is capable of accurately identifying hazardous wildlife species, including lions, leopards, monkeys, and elephants. The implementation on an edge device ensures low-latency operation, reliability in remote environments, and enhanced data privacy. A user-friendly dashboard was also developed to provide live monitoring and instant alerts, improving situational awareness and

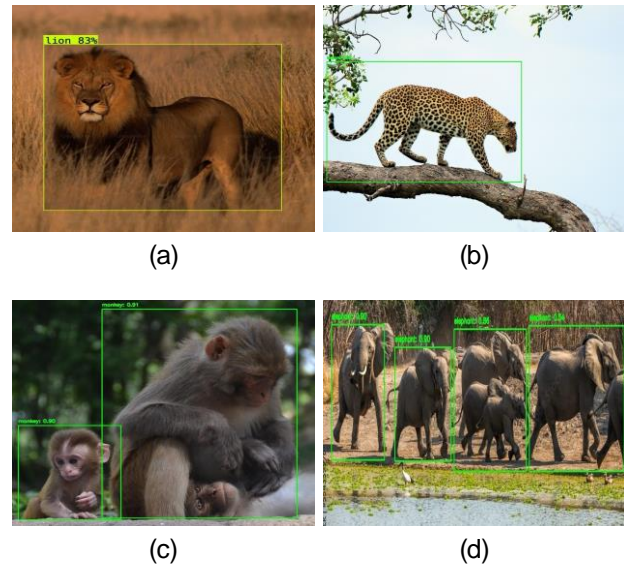


Fig. 3. Sample detection results demonstrating successful identification and localization of (a) Lion, (b) Leopard, (c) Monkey, and (d) Elephant.

response times. The system's modular design allows users to extend its functionality by adding new animal classes according to specific requirements. Overall, the solution demonstrates a cost-effective, scalable, and efficient method for strengthening safety measures in wildlife-prone regions. Future work will focus on expanding the dataset, optimizing detection under varying environmental conditions, and integrating additional communication modules for broader alert dissemination.

REFERENCES

- [1] M. A. Tabak et al. Animal Detection and Classification from Camera Trap Images Using Different Mainstream Object Detection Architectures. *Animals*, 12(15):1982, August 2022.
- [2] M. Ve'lez-Reyes et al. Animal Species Recognition with Deep Convolutional Neural Networks from Ecological Camera Trap Images. *Animals*, 13(9):1526, May 2023.
- [3] S. Kumar et al. YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n. *Ecological Informatics*, 82:102745, July 2024.
- [4] Y. Wang, J. Li, and X. Zhang. Nighttime wildlife object detection based on YOLOv8-night. *Electronics Letters*, 60(15):e13305, August 2024.
- [5] M. Chen, L. Wang, and H. Liu. WildARE-YOLO: A lightweight and efficient wild animal recognition model. *Ecological Informatics*, 80:102539, April 2024.
- [6] Geethanjali P., Metun Nivin, and M. Rajeswari. Advances in ecological surveillance: Real-time wildlife detection using mobilenet-ssd v2 cnn machine learning. 11 2023.
- [7] Emrah S, ims,ek, Baris, O' zyer, Levent Bayındır, and Gu'ls,ah Tu'mu'klu' O' zyer. Human-Animal Recognition in Camera Trap Images. In 2018 26th Signal Processing and Communications Applications Conference (SIU), pages 1–4, 2018.
- [8] Mohamed Fathy, Mohamed K. Elhadad, and Mohamed A. Elshafey. Comprehensive Performance Evaluation of YOLOv8-11 Models for Object Recognition in Remote Sensing Imagery. In 2025 15th International Conference on Electrical Engineering (ICEENG), pages 1–6, 2025.
- [9] Rodolfo Bonnin, Claudio Delrieux, and Mar'ia Fabiana Piccoli. Deep Learning-Based Visual Aid for Low Vision. *IEEE Embedded Systems Letters*, pages 1–1, 2025.
- [10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

- [11] Cheng-Kai Lu, Jun-Yu Shen, Cheng-Hung Lin, Chung-Yueh Lien, and Ding Su Yen. Efficient Embedded System for Small Object Detection: A Case Study on Floating Debris in Environmental Monitoring. *IEEE Embedded Systems Letters*, 17(4):264–267, 2025.
- [12] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023.
- [13] A. G. Howard et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, April 2017.
- [14] T. Nguyen et al. Wildlife Real-Time Detection in Complex Forest Scenes Based on YOLOv5s Deep Learning Network. *Remote Sensing*, 16(8):1350, April 2024.