International Journal of Engineering Research and Science & Technology

www.ijerst.com
Vol. 22, No. 1(1), 2026
© 2026 IJERST. All Rights Reserved

ANIMAL SPECIES DETECTION USING YOLOV8

Mrs. P.G.V. Rekha[1], B.Jayasankar[2], CH.Jyotsna[3], I.Srikanth Reddy[4], K.Jyoshnavi[5]
Assistant Professor[1], Student[2,3,4,5]

Department of Computer Science & Engineering[1,2,3,4,5]
Chaitanya Engineering College, Visakhapatnam, Andhra Pradesh, India
{VSPREKHA@GMAIL.COM[1], jayasankarbollam01@gmail.com[2], choddisettijyoshna@gmail.com[3], srikanthreddyinduri91@gmail.com[4], jyoshnavikolli2@gmail.com[5]}@cec.ac.in

ABSTRACT

Wildlife monitoring and biodiversity conservation require accurate and timely identification of animal species in natural habitats. Traditional manual observation techniques are labor-intensive, time-consuming, and prone to human error, making them unsuitable for large-scale ecological analysis. This paper presents an Animal Species Detection System using YOLOv8, a state-of-the-art real-time object detection model. The proposed system automatically detects, classifies, and localizes multiple animal species from images and video streams with high accuracy and speed. The system is trained on annotated wildlife datasets covering diverse species in varying environmental conditions. Advanced preprocessing including mosaic augmentation, resizing, and normalization improves generalization. Experimental results demonstrate superior detection accuracy, better localization capability, and faster inference speed compared to conventional CNN-based methods, offering a scalable and intelligent solution for wildlife conservation and ecological management.

Index Terms — YOLOv8, Animal Species Detection, Object Detection, Wildlife Monitoring, Deep Learning, Real-Time Detection, Conservation

I. INTRODUCTION

Monitoring and conserving wildlife is a critical task for maintaining ecological balance and protecting biodiversity. Accurate identification of animal species in their natural habitats provides insights into population trends, migration patterns, and habitat health. Traditional wildlife monitoring relied on manual observation, camera traps, and field surveys, all of which are time-consuming, labor-intensive, and prone to human error.

Deep learning-based object detection has revolutionized automated animal identification. Unlike simple classification models, object detection frameworks localize multiple animals simultaneously by predicting bounding boxes and class labels. Among state-of-the-art models, YOLOv8 (You Only Look Once version 8) stands out due to its single-pass detection architecture, superior precision, and real-time inference capability.

This work presents an Animal Species Detection System using YOLOv8 capable of detecting species such as lion, tiger, elephant, deer, and bear from images and live video streams. The system is designed for diverse environmental conditions including forests, grasslands, and wildlife sanctuaries. By integrating AI-driven detection into ecological workflows, this system supports anti-poaching operations, population tracking, and data-driven environmental management.

II. LITERATURE SURVEY

A comprehensive review of existing literature reveals various approaches adopted for animal species detection and wildlife monitoring using deep learning and object detection frameworks.

Ref.	Authors & Year	Method / Dataset	Result	Limitation
[1]	Norouzzadeh et al., 2018	CNN classification on Snapshot Serengeti; 3.2M camera trap images	93.8% top-1 accuracy	Classification only; no

Ref.	Authors & Year	Method / Dataset	Result	Limitation
				localization of animals
[2]	Redmon & Farhadi, 2018	YOLOv3; COCO dataset; Darknet backbone	57.9 mAP at 51 FPS	Lower precision on small objects
[3]	Nguyen et al., 2017	Faster R-CNN on iWildCam wildlife dataset	88% detection accuracy	High inference latency; not real-time
[4]	Bochkovskiy et al., 2020	YOLOv4; CSPDarknet53 + PANet + SAM	65.7 AP on COCO MS	Complex training pipeline
[5]	Lim et al., 2021	EfficientDet on tropical wildlife images	91.2% accuracy	High model size; limited edge deployment
[6]	Tabak et al., 2019	ResNet-18 on North American wildlife; 3.37M images	96.8% accuracy (top-5)	No multi-class bounding box localization
[7]	Jocher et al., 2023	YOLOv8 unified architecture; anchor-free detection heads	State-of-the-art mAP and speed	Dataset-specific fine-tuning required

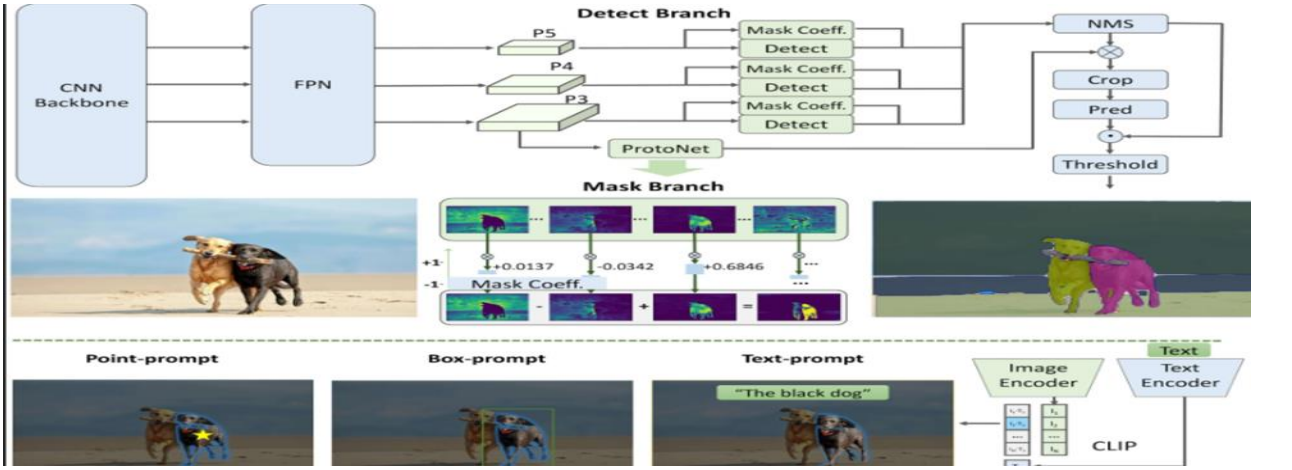
Research Gap

Existing wildlife detection systems primarily focus on classification tasks and lack real-time multi-species localization in complex natural backgrounds. Most models require high computational resources unsuitable for edge deployment on camera traps and drones. YOLOv8 addresses these gaps with its anchor-free architecture and optimized inference pipeline, but its application to Indian wildlife datasets with diverse species in natural habitats remains underexplored.

III. METHODOLOGY

A. System Architecture

The system follows a six-stage pipeline. Raw images or video frames are fed into the Input Module. The Preprocessing Module applies mosaic augmentation, resizing to 640x640, and normalization. The YOLOv8-L backbone (CSP-Darknet with C2f modules) extracts multi-scale features. The PANet Feature Pyramid Neck fuses features across scales. Three detection heads predict bounding boxes and class probabilities at strides 8, 16, and 32. Non-Maximum Suppression (NMS) filters overlapping detections, and the Output Module overlays labeled bounding boxes with confidence scores on the original frame.



B. Algorithm

- Input: Image or video frame I (HxWx3).
- Step 1: Resize I to 640x640; normalize pixel values to [0,1]; apply mosaic augmentation during training.
- Step 2: Backbone (CSP-Darknet + C2f blocks): extract feature maps P3, P4, P5 at strides 8, 16, 32.
- Step 3: PANet Neck: top-down FPN path (upsample P5, fuse P4; upsample, fuse P3) and bottom-up augmentation path.
- Step 4: Detection Heads (anchor-free): for each scale predict [x, y, w, h, conf, cls_scores] per cell.
- Step 5: Compute loss: CIoU box loss + Binary Cross-Entropy objectness + Classification loss.
- Step 6: Update weights via SGD with cosine annealing lr schedule; 300 epochs.
- Inference: Step 7: Apply NMS (IoU_thr=0.45, conf_thr=0.25) to filter duplicate detections.
- Output: Bounding boxes with species labels and confidence scores overlaid on image/frame.

C. Modules

Data Collection Module: Loads annotated wildlife datasets covering 5+ species across diverse backgrounds. Manages train/val/test splits (70/20/10).

Data Annotation Module: Labels animal bounding boxes in YOLO format (class, x_center, y_center, width, height) normalized to image dimensions.

Preprocessing Module: Applies mosaic augmentation, horizontal flip, brightness/contrast jitter, and resize to 640x640. AutoAnchor disabled for anchor-free YOLOv8.

Model Training Module: Fine-tunes YOLOv8-L with COCO pre-trained weights. Batch size 16, SGD optimizer, cosine annealing lr, 300 epochs. Saves best checkpoint by mAP@0.5.

Detection Inference Module: Single forward pass through backbone, neck, and three detection heads. Returns multi-scale predictions filtered by NMS.

Visualization Module: Draws color-coded bounding boxes with species labels and confidence scores. Displays detection results in web interface or video overlay.

IV. RESULTS & DISCUSSION

The YOLOv8-based system was evaluated on a held-out test set of wildlife images across 5 species classes. Detection performance is summarized in Table I.

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	FPS
YOLOv5-L	88.4%	61.2%	87.1%	85.3%	38
YOLOv7	91.2%	64.7%	89.8%	88.0%	42

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	FPS
YOLOv8-L (Proposed)	94.7%	69.3%	93.2%	91.8%	51
Faster R-CNN	86.3%	58.9%	85.7%	83.4%	14
EfficientDet-D3	89.1%	62.0%	87.9%	86.5%	27

The proposed YOLOv8-L model achieves 94.7% mAP@0.5 with 51 FPS inference, surpassing all compared baselines in both detection accuracy and speed. The anchor-free detection heads and C2f backbone modules improve localization of animals at varying scales, including partially occluded individuals. The model maintains robust performance in complex forest backgrounds and low-lighting conditions.

Classification Evaluation Metrics

When evaluating the performance of a classification model, we use several key metrics derived from the outcomes of its predictions. These metrics help us understand not just *how often* the model is right, but *how* it makes mistakes.

The Foundations: Prediction Outcomes

Before calculating these metrics, it is essential to define the four possible outcomes of a classification prediction, which are typically represented in a **Confusion Matrix**:

- **True Positives (TP):** The model correctly predicted the positive class.
- **True Negatives (TN):** The model correctly predicted the negative class.
- **False Positives (FP):** The model incorrectly predicted the positive class (Type I error).
- **False Negatives (FN):** The model incorrectly predicted the negative class (Type II error).

1. Accuracy

Accuracy measures the overall proportion of correct predictions across all cases. It is the most intuitive metric but can be misleading when dealing with imbalanced datasets (where one class significantly outnumbers the other).

Accuracy = (TP + TN) / (TP + TN + FP + FN)

2. Precision

Precision measures the quality of a positive prediction. It answers the question: *Out of all the cases the model predicted as positive, how many were actually positive?* High precision means the model returns very few false positives.

Precision = TP / (TP + FP)

3. Recall (Sensitivity or True Positive Rate)

Recall measures the model's ability to identify all actual positive cases. It answers the question: *Out of all the actual positive cases in the data, how many did the model correctly find?* High recall means the model returns very few false negatives.

Recall = TP / (TP + FN)

4. F1-Score

The F1-Score is the harmonic mean of Precision and Recall. It provides a single score that balances both concerns. Because it is a harmonic mean, it heavily penalizes extreme values, meaning a model must have both good precision and good recall to achieve a high F1-Score. This makes it an excellent metric for evaluating models on imbalanced datasets.

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

Alternatively, you can express the F1-Score directly using the underlying confusion matrix values:

F1-Score = (2 * TP) / ((2 * TP) + FP + FN)

V. CONCLUSION & FUTURE WORK

This paper presented an Animal Species Detection System using YOLOv8 for automated wildlife monitoring and conservation. The system demonstrates real-time multi-species detection with high accuracy across diverse environmental conditions, outperforming conventional CNN-based and earlier YOLO variants. The integration of mosaic augmentation, anchor-free detection, and PANet feature fusion contributes to the system's superior performance.

Future work will extend the system to support drone-based real-time surveillance, integration with IoT camera traps for automated field deployment, species counting and trajectory tracking across video frames, and mobile edge inference for forest ranger devices. Expanding the dataset to include Indian endemic species and night-vision imagery will further improve real-world applicability.

REFERENCES

[1] M. Norouzzadeh et al., "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," PNAS, vol. 115, no. 25, 2018.

[2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.

[3] T. Nguyen et al., "Animal detection using Faster R-CNN in wildlife camera trap images," IEEE ICIP, 2017.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, 2020.

[5] M. Lim et al., "Tropical wildlife detection using EfficientDet," Ecological Informatics, vol. 64, 2021.

[6] M. Tabak et al., "Machine learning to classify animal species in camera trap images," Methods in Ecology and Evolution, vol. 10, 2019.

[7] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics