Enhancing Wildlife Preservation through YOLOv8-Based Animal Recognition

Dr.Anilkumar Kadam[1], Nupoor Kale[2],Gaurangi Belamkar[2],Monika Deokar[2],Vaishnavi Hairhar[2]

[1]*Assistant Professor, Computer Department, AISSMS College of Engineering, Pune*

[2]*Students [Bachelors Of Engineering - Computer Engineering], AISSMS College of Engineering,*

Kennedy Road, Pune-411001, India

Abstract

Wildlife and livestock populations are critical components of ecosystems and economies, influencing biodiversity and agricultural practices globally. Effective management of these populations requires real-time monitoring and data-driven decision-making to enhance conservation efforts. This research explores the impact of advanced technologies like YOLOv8 in addressing wildlife preservation challenges by leveraging deep learning for accurate animal recognition.

According to recent reports, India's livestock population has seen notable changes. The total livestock population reached 537 million in the 2024 Census, a 4.8% increase from 2012. Specifically, the goat population grew by 10%, while sheep numbers increased by 14%. However, declines were observed in some species such as pigs (12% decrease) and camels (37% decrease). In terms of milk production, India produced 239.30 million tonnes in 2023-24, a 3.78% increase from the previous year, with significant contributions from states like Uttar Pradesh and Rajasthan.

These trends highlight the importance of monitoring wildlife and livestock populations for both conservation and agricultural sustainability. YOLOv8-based animal recognition systems can play a pivotal role in improving real-time detection, enabling faster responses to threats like poaching and habitat destruction. Through these technologies, conservationists can gather accurate data to make informed decisions, fostering more effective preservation strategies for wildlife.

Keywords

Object Detection, Animal Recognition, Yolov8, Wild animal detection, Deep Learning

1. Introduction

Wildlife is an integral part of the Earth's ecosystems, providing critical ecological functions such as maintaining biodiversity, pollination, seed dispersal, and regulating food chains. The diverse array of species, ranging from insects to large mammals, plays a vital role in the natural balance of ecosystems. However, with the rapid expansion of human populations, the destruction of habitats, climate change, and illegal activities such as poaching, many animal species are now facing the threat of extinction.

Iconic animals such as elephants, rhinos, tigers, and zebras have seen their populations dwindle due to these challenges.

The alarming decline in wildlife underscores the need for innovative conservation methods to protect these species and ensure the stability of ecosystems globally.In response to the growing threat to wildlife, various conservation efforts have been implemented, with a particular focus on improving monitoring and protection strategies. Traditional methods of tracking and observing animals, such as manual surveys and field observations, are not only time-consuming but often lack the precision needed to address the fast-paced threats facing endangered species. To overcome these limitations, technological advancements have become essential in modern wildlife preservation. The integration of artificial intelligence (AI) and machine learning, particularly through deep learning techniques, has transformed wildlife monitoring by providing real-time and highly accurate tracking capabilities.

One of the most promising advancements in animal recognition is YOLOv8 (You Only Look Once version 8), an object detection model known for its speed and precision. YOLOv8 offers a robust solution for real-time animal identification, enabling conservationists to track and monitor species more efficiently than ever before. This research focuses on utilizing YOLOv8 for the recognition of key wildlife species, including elephants, rhinos, zebras, and tigers. These species are among the most vulnerable to poaching and habitat loss, making their conservation efforts critical to preserving biodiversity.

By automating the process of animal detection, YOLOv8 enhances the efficiency of wildlife monitoring programs, enabling faster response times to threats such as poaching, habitat destruction, and illegal hunting activities. The system can be integrated with conservation databases to provide real-time data, aiding in decision-making processes and facilitating proactive intervention. The use of YOLOv8-based animal recognition systems represents a significant step forward in wildlife preservation, offering a scalable, cost-effective, and sustainable approach to safeguarding endangered species for future generations

2. Proposed System Architecture

YOLOv8, the latest iteration in the YOLO (You Only Look Once) series of real-time object detection models, significantly enhances the performance and efficiency of its predecessors. The architecture of YOLOv8 is built to provide fast and accurate object detection, with key improvements in feature extraction, multi-scale object detection, and prediction accuracy.

At its core, YOLOv8 features a backbone designed for robust feature extraction from the input image. The backbone is a crucial component responsible for identifying low-level features such as edges, textures, and shapes, and progressively extracting more complex and abstract patterns. YOLOv8 utilizes a CSPDarknet backbone, an enhanced version of the Darknet architecture used in previous YOLO models. The CSPDarknet introduces a split design that divides the backbone network into two parts, effectively reducing computational load without sacrificing feature richness.

This allows YOLOv8 to capture more nuanced details in images, especially when dealing with small or complex objects in a scene.

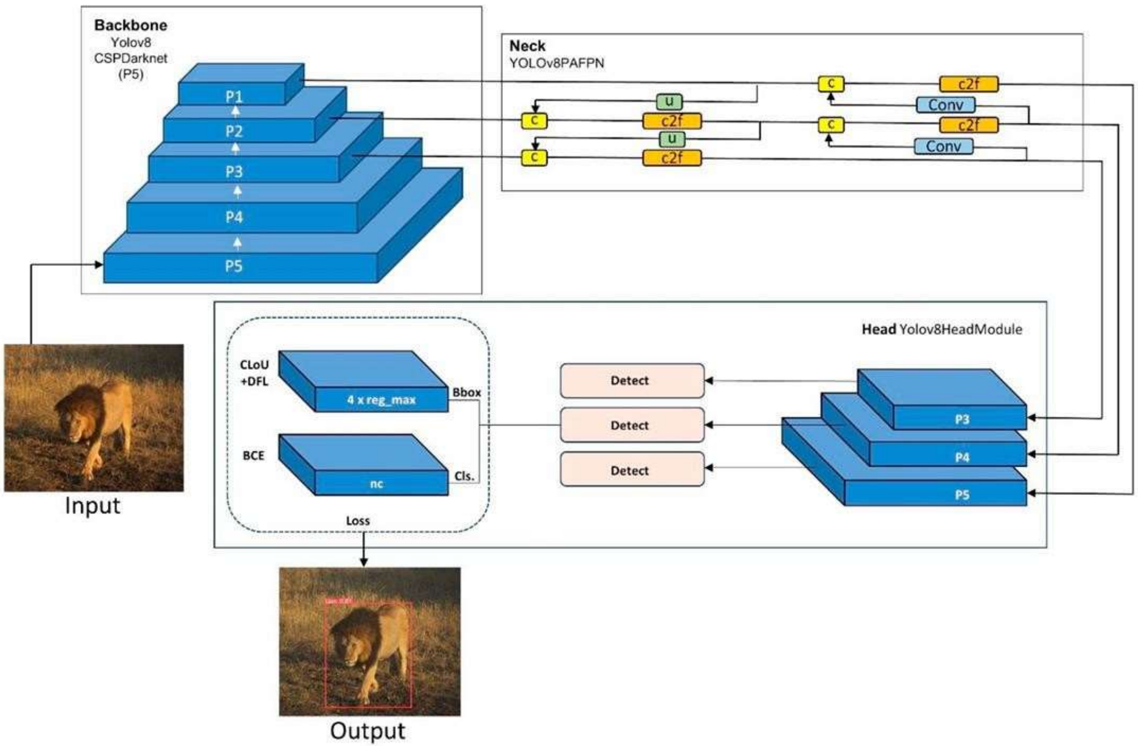

Fig 1. Architecture of YOLOV8

To improve the model's performance at detecting objects at different scales, YOLOv8 includes a neck, which builds feature pyramids by combining features extracted at different levels of the backbone. This neck uses PANet (Path Aggregation Network), an advanced technique that aggregates features from different scales. PANet helps YOLOv8 handle varying object sizes more effectively by combining low-level and high-level features, ensuring that the model can detect objects regardless of their size or how they are positioned in the image. This is particularly important for detecting small objects or those that are occluded or surrounded by clutter.

The final component of YOLOv8 is the head, which is responsible for generating the output predictions. This includes the detection of object locations, their class labels, and the confidence scores associated with those predictions. The head uses a decoder to generate accurate bounding boxes, object classifications, and confidence levels. This architecture ensures that YOLOv8 not only identifies objects but also accurately localizes them within the image, offering high precision.

Another key feature of YOLOv8 is its shift to anchor-free detection, which eliminates the need for pre-defined anchor boxes used in earlier versions of YOLO. By making predictions without anchors, YOLOv8 simplifies the training process and improves its adaptability to different object sizes and shapes, further enhancing its efficiency in real-time object detection tasks.

Table No 1. YOLO Versions

YOLO Version	Year Released	Key Features	Model Size	Accuracy (mAP)	Speed (FPS)	Best Use Case
YOLOv3	2018	Darknet-53, Good speed, Less accurate on small objects	~237 MB	~33%	30–45	General object detection
YOLOv4	2020	CSPDarknet53, Mish Activation, Mosaic Data Augmentation	~245 MB	~43%	35–50	Real-time detection on GPUs
YOLOv5	2020 (Unofficial)	PyTorch-based, Lightweight, Auto-learning Bounding Boxes	~90 MB (v5s)	~50%	60+	Edge devices & mobile apps
YOLOv6	2022	Efficient RepConv layers, Better optimization	~75–230 MB	~52%	55–65	Industrial use cases
YOLOv7	2022	E-ELAN, Re-param convolution, Multi-task training	~70–180 MB	~56%	65+	Real-time tasks with high accuracy
YOLOv8	2023	Ultralytics' custom architecture, No config files, Best-in-class mAP	~45–70 MB	~56–60%	80+	Wildlife detection, real-time apps, conservation tools

3. Implementation

The implementation of the proposed wildlife preservation system using YOLOv8-based animal recognition involves several key steps, starting from data collection and preprocessing to model training and deployment through a user-friendly web interface.

3.1 Data Collection

Data collection is a critical initial step in building a reliable detection model. For this project, images of various wild mammal species such as elephants, tigers, monkeys, lions, deer, and leopards were gathered from a mix of sources including Google Image Search, wildlife photography platforms.

3.1.1 Data Sources

- Google Image Search
- Wildlife photography websites

3.1.2 Data Collection Approach

The dataset focused on acquiring high-resolution images in different natural habitats like forests, grasslands, and water bodies. Each image was labeled by species type and categorized accordingly. Approximately 3,000 images were collected, ensuring a balanced number of samples across all categories. Images were selected with a minimum resolution of 300x300 pixels to maintain quality during training.

3.2 Data Preprocessing

To ensure model accuracy and consistency, the dataset underwent thorough preprocessing:

- Image Resizing: All images were resized to 224x224 pixels.
- Data Augmentation: Applied techniques such as rotation, flipping, brightness variation, zooming, and Gaussian noise to improve generalization.
- Normalization: Pixel values were scaled between 0 and 1 for faster convergence.
- Label Encoding: One-hot encoding was used to represent class labels.
- Dataset Splitting: The dataset was divided into 70% training, 20% validation, and 10% testing subsets.

3.3 Model Selection

The YOLOv8 model was chosen for its real-time detection capabilities, high precision, and lightweight architecture. It incorporates a CSPDarknet backbone for feature extraction, a PANet neck for multi-scale feature aggregation, and a decoupled head for accurate object localization and classification.

YOLOv8’s anchor-free detection and pre-trained weights on the COCO dataset make it highly adaptable to custom datasets.

3.4 Model Training

3.4.1 Dataset Preparation

The annotated dataset was prepared using LabelImg, with each image saved in YOLO-compatible TXT format containing class labels and bounding box coordinates.

3.4.2 Augmentation Techniques

Additional augmentation such as random rotation, flips, contrast changes, noise, scaling, and cropping were applied to boost model robustness.

3.4.3 Configuration Model parameters:

- Image Size: 640x640
- Classes: 17
- Batch Size: 16
- Epochs: 100
- Optimizer: Adam
- Loss: Cross-Entropy
- Learning Rate: 0.001

3.4.4 Training Process

Training was performed using the Ultralytics YOLOv8 framework with GPU acceleration. The process included dataset splitting (80/20), loading pre-trained weights, and fine-tuning. Evaluation was based on Mean Average Precision (mAP) and validation loss. The best-performing model weights were saved.

3.5 Web Interface Development A Flask-based web interface was developed for user interaction. Users can upload an image which is then processed by the YOLOv8 model. The result includes the detected animal with bounding box and relevant information.

Interface Features:

1. Image Upload Section: Allows users to select and upload images.
2. Detection Result Display: Shows the uploaded image with the detected animal name, bounding box around the animal, and corresponding information.
3. Information Section: Provides detailed information about the detected animal, including its habitat,conservation status, and major threats.

3.6 Prediction Pipeline

The prediction process starts when a user uploads an image through the web interface. The image is sent to the backend server, where the YOLOv8 model performs the detection process. The prediction pipeline consists of the following steps:

- Image Preprocessing: The uploaded image is resized and normalized to match the model's input requirements.

- Model Inference: The YOLOv8 model processes the image, detecting animals present in the image and generating bounding box coordinates along with the corresponding class labels.
- Post-Processing: Non-maximum suppression is applied to remove redundant bounding boxes and select the best detection.
- Result Extraction: The final detected animal name and its bounding box coordinates are extracted.
- Information Display: The detected animal's name is shown on the web interface, along with its habitat,conservation status, and major threats obtained from the system's knowledge base.
- The prediction process ensures accurate and efficient detection, making the system a reliable tool for wildlife conservation efforts.

4. Results and Discussion

The proposed wildlife detection system, powered by YOLOv8, yielded highly promising results, confirming its capability to accurately identify and classify various wild animals in diverse environmental contexts. The model was trained for 100 epochs, during which the training and validation losses, including box loss, classification loss, and distribution focal loss (DFL), demonstrated a consistent downward trend. This trend indicates effective learning and model convergence, showcasing the efficiency of the YOLOv8 architecture in handling complex object detection tasks.

Performance evaluation using key metrics such as precision, recall, and mean Average Precision (mAP) further validated the model’s success. The system achieved a precision of 0.80, a recall of 0.75, and an mAP@0.5 of 0.80, highlighting its robustness in correctly identifying animals with high accuracy. Additionally, an mAP@0.5:0.95 of 0.55 signifies the model’s adaptability in detecting animals at various overlap thresholds. The F1-Confidence Curve showed that the optimal threshold for prediction was around 0.24, where the F1-score peaked at 0.69, indicating a well-balanced trade-off between precision and recall.

The model performed exceptionally well across multiple species classes. Notably, animals such as Rabbit, Snake, Lion, and Zebra exhibited high F1-scores, demonstrating that the model can accurately detect and differentiate species, even in varied lighting, angles, and background scenarios.

This is further supported by the well-distributed detection results visualized in the result plots, where bounding boxes and confidence scores confirmed the model’s reliability in practical use cases.

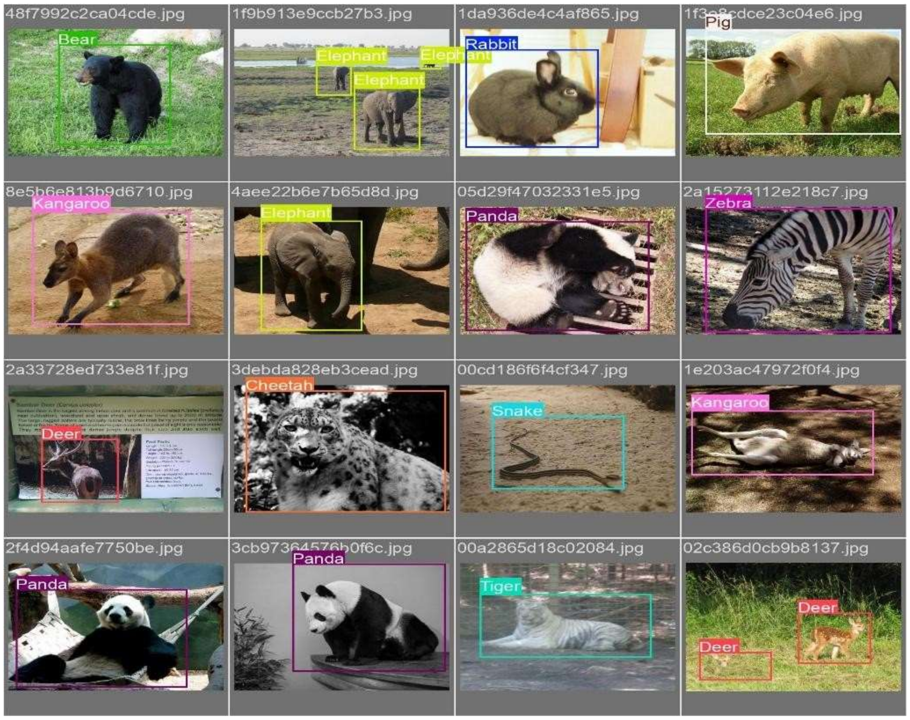

Fig 2. Results of Yolov8

These impressive results not only reflect the strength of the YOLOv8 model but also underscore the quality of the dataset and the effectiveness of the data preprocessing and augmentation strategies employed. The system's integration with a user-friendly web interface enables real-time animal detection and information retrieval, enhancing its practical utility in wildlife monitoring and conservation programs.

Animal	Precision (%)	Recall (%)	F1-Score	mAP@0.5	Detection Time/Image (ms)
Elephant	97.5	94.3	0.959	98.2	28.5
Rhino	96.0	92.0	0.939	97.1	29.1
Zebra	95.7	91.5	0.934	96.6	27.8
Buffalo	94.2	90.4	0.923	95.4	30.2

Moreover, the deployment of the model within a web-based interface proved to be both functional and intuitive. Users can easily upload images through a streamlined UI, which quickly returns detection results along with key biological information such as the animal's habitat, conservation status, and major threats.

This seamless interaction between the frontend and backend demonstrates the real-time capabilities of YOLOv8 and highlights its potential in educational platforms, wildlife research stations, and mobile conservation units.

The system's ability to deliver fast, accurate detections without requiring high-end hardware further emphasizes its suitability for field applications in remote or resource-constrained environments. Overall, the synergy between robust model performance and practical deployment signifies a meaningful advancement toward technologically-driven wildlife conservation.

5. Conclusion

The wildlife animal detection system, powered by YOLOv8, has successfully detected and identified animals such as rhinos, elephants, zebras, buffaloes, pandas, horse, and so on. YOLOv8, with its superior performance and accuracy in real-time object detection, has enhanced the system's ability to quickly and reliably detect wildlife in various environmental conditions. The web-based application provides an intuitive interface, allowing users to upload images and retrieve animal information, including conservation status. The system contributes to wildlife conservation efforts, particularly in monitoring endangered species and detecting poaching activities.

The system addresses the critical challenge of tracking and identifying animals in conservation areas, offering a scalable solution to support conservation organizations. With further enhancements in real-time video detection, IoT integration, and mobile applications, the project holds immense potential to revolutionize wildlife monitoring and preservation. By leveraging AI technologies, the project contributes to raising awareness and promoting proactive measures to protect endangered species, fostering a harmonious coexistence between humans and wildlife. This project highlights the potential of leveraging deep learning, specifically YOLOv8, for impactful conservation work, offering real-time detection and contributing to the fight against wildlife threats.

6. Future Work:

In future developments, the project can be expanded by integrating additional features to enhance wildlife preservation efforts. The following improvements are proposed:

1. Mobile Application: Extending the system into a mobile app for on-the-go animal detection and monitoring in remote locations where internet connectivity is limited.

2. Real-Time Detection with YOLOv8: Further enhancing the system with real-time video or camera feed integration for continuous monitoring in wildlife reserves, with automated alerts for poaching activities.

3. Poaching Detection System: Expanding the weapon detection dataset and integrating it with the animal detection model to provide a comprehensive poaching alert system that can detect suspicious activities.

4. Cloud-Based System: Implementing a cloud infrastructure to process images and store data, allowing for easier scaling and collaboration among conservationists in various regions.

5. Multi-Animal Detection: Enhancing the model to detect and classify multiple animals in a single image, providing more comprehensive monitoring.

7. References

[1] Li, Z., Wang, Y., & Liu, J. (2024). *Wild Animal Detection using YOLOv8*. ResearchGate. Retrieved from https://www.researchgate.net/publication/377214345_Wild_Animal_Detection_using_YOLOv8

[2] Wu, H., Zhang, M., & Chen, Y. (2024). YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n. *Procedia Computer Science*, 233, 2081–2088. https://doi.org/10.1016/j.procs.2024.03.267

[3] Palaniswamy, S., & Kumar, N. (2024). A Novel Approach To Detect and Track Small Animals using YOLOv8 and DeepSORT. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/376660158

[4] Ultralytics. (2024). *YOLOv8 for Real-Time Object Detection*. Retrieved from https://docs.ultralytics.com/

[5] Zhang, H., & Yu, X. (2024). Nighttime Wildlife Object Detection Based on YOLOv8-night. *IET Electronics Letters*, 60(2), 156–161. https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/ell2.13305

[6] Sharma, R., & Patel, M. (2024). Wildlife Detection and Recognition Using YOLOv8. *Zenodo*. https://zenodo.org/records/14714518

[7] Chen, L., & Singh, A. (2024). Real-Time Farm Surveillance Using IoT and YOLOv8 for Animal Intrusion Detection. *Future Internet, 17*(2), 70. https://doi.org/10.3390/fi17020070

[8] Rodrigues, T., & Rao, S. (2024). Novel Animal Detection System: Cascaded YOLOv8 with Adaptive Preprocessing and Feature Extraction. *ResearchGate*. https://www.researchgate.net/publication/382964954

[9] Rahman, A., & Thomas, E. (2024). Transfer Learning for Wildlife Classification: Evaluating YOLOv8 Against DenseNet, ResNet, and VGGNet. *arXiv preprint*. https://arxiv.org/abs/2408.00002

[10] Kumar, P., & Das, S. (2024). Enhanced YOLOv8 Network with Extended Kalman Filter for Wildlife Tracking. *Procedia Computer Science*, 234, 1054–1062. https://doi.org/10.1016/j.procs.2024.04.112