International Journal on Emerging Research Areas (IJERA)

DOI: 10.5281/zenodo.14714518 *ISSN : 2230-9993*

Wildlife Detection And Recognition Using YOLO V8

Ms.Shiney Thomas
Dept. of Computer Science
Amal Jyothi College of Engineering
(Autonomous)
Kottayam, India
shineythomas@amaljyothi.ac.in

Elsa George
Dept. of Computer Science
Amal Jyothi College of Engineering
(Autonomous)
Kottayam, India
elsageorge2025@cs.ajce.in

Alphonsa Francis
Dept. of Computer Science
Amal Jyothi College of Engineering
(Autonomous)
Kottayam, India
alphonsafrancis2025@cs.ajce.in

Anna Job
Dept. of Computer Science
Amal Jyothi College of Engineering
(Autonomous)
Kottayam, India
annajob2025@cs.ajce.in

Ann Maria James
Dept. of Computer Science
Amal Jyothi College of Engineering
(Autonomous)
Kottayam, India
annmariajames2025@cs.ajce.in

***Abstract*—The use of YOLOv8 for wildlife detection and recognition has transformed real-time monitoring across diverse environments, particularly in rural, forested, and human-wildlife conflict zones. Its lightweight architecture, efficient feature extraction, and deep learning capabilities make it a preferred tool for wildlife conservation. YOLOv8's ability to detect and classify animals in real-time has enhanced wildlife population monitoring, reduced risks of human-wildlife encounters, and contributed to biodiversity conservation.**

A major advancement in YOLOv8 is its ability to perform well under low-visibility conditions, such as nighttime, fog, and haze. These scenarios traditionally present significant c hallenges for detection models, as poor lighting and environmental interference can obscure critical visual features. However, YOLOv8, along with enhanced models like YOLO-SAG and WL-YOLO, addresses these issues by incorporating attention mechanisms, adaptive preprocessing, and lightweight modules. This allows the models to maintain high detection accuracy, often exceeding 97

Nighttime detection has been significantly improved by integrating glow reduction and adaptive preprocessing techniques, which handle artificial lighting, light scattering, and low contrast—issues that typically hinder detection in nocturnal settings. As a result, YOLOv8 and similar models offer robust and accurate detection in dimly lit environments.

These enhancements in YOLOv8-based models provide a balance between accuracy, speed, and computational efficiency, reducing false positives and increasing reliability in real-time applications. With its ability to handle low visibility and complex environments, YOLOv8 is a crucial tool for wildlife conservationists, supporting real-time monitoring, behavior analysis, and rapid response to human-wildlife conflicts

***Index Terms*—Wildlife Detection, YOLOv8, Object Detection, Nighttime Detection, and Deep Learning**

I. Introduction

Wildlife detection and monitoring have become critical components of biodiversity conservation and management, especially in light of increasing human encroachment into natural habitats. As human populations expand, interactions between wildlife and human activities have escalated, leading to conflicts that threaten both safety and wildlife preservation. Effective detection systems are essential for minimizing human-wildlife conflicts, protecting endangered species, and ensuring the sustainability of ecosystems. The application of advanced object detection technologies, particularly deep learning models like YOLOv8, has revolutionized the field of wildlife monitoring by enhancing the accuracy and speed of wildlife detection in various environments.

Traditional wildlife monitoring methods, such as manual surveys and camera traps, have limitations in terms of labor intensity and data processing efficiency. Although camera traps are effective for capturing wildlife images, the subsequent identification and classification often rely on manual labor, which can be time-consuming and expensive . To address these challenges, researchers have increasingly turned to deep learning techniques that automate the detection process and enable real-time analysis of wildlife populations. The integration of models like YOLO (You Only Look Once) has significantly improved detection accuracy, enabling researchers to identify and classify various species with minimal human intervention .

Despite advancements in deep learning models, wildlife detection remains fraught with challenges. Poor visibility conditions, such as nighttime, fog, and haze, can significantly degrade detection performance . Traditional models often struggle to accurately detect wildlife in these adverse conditions due to limitations in feature extraction and image quality. Recent studies have focused on enhancing models like YOLOv8 to improve their robustness under low-visibility scenarios, employing techniques such as dehazing and adaptive preprocessing to address issues like light scattering and contrast reduction .

Incorporating attention mechanisms and lightweight architectures, recent adaptations of YOLO, such as YOLO-SAG and WL-YOLO, have demonstrated remarkable improvements

DOI:10.5281/zenodo.14714518 *ISSN : 2230-9993*

in accuracy and speed, making them suitable for real-time applications in wildlife monitoring . Additionally, the advent of dehazing algorithms allows for better image quality in challenging weather conditions, improving the overall reliability of wildlife detection systems .

As the demand for effective wildlife monitoring continues to grow, the combination of advanced detection algorithms with innovative preprocessing techniques presents a promising avenue for future research and application. This paper aims to explore the capabilities of YOLOv8 and its variants in wildlife detection, particularly under adverse environmental conditions, and assess their effectiveness in supporting conservation efforts and mitigating human-wildlife conflicts.

II. Advancements in Animal Detection Models

The introduction of Convolutional Neural Networks (CNNs) marked a pivotal shift in object detection. Early models such as AlexNet and VGGNet automated feature extraction, significantly improving accuracy. However, these models were computationally intensive and not suitable for real-time applications. Despite their limitations, they laid the groundwork for more advanced architectures.

A major leap occurred with the development of Faster R-CNN and YOLO (You Only Look Once). Faster R-CNN employed a two-stage detection process, generating region proposals before classification, which resulted in high accuracy but slower inference times. In contrast, YOLO introduced a single-shot detection approach, treating the detection task as a regression problem. This innovation enabled real-time applications with a balanced trade-off between speed and precision, making models like YOLOv3 highly effective in diverse wildlife environments.

Further advancements emerged with YOLOv5, which optimized detection speed, accuracy, and computational efficiency, making it ideal for real-time monitoring. Specialized models like Dehazing YOLO tackled specific challenges, such as low-visibility detection in foggy or misty conditions, enhancing the reliability of wildlife detection systems.

Most recently, YOLOv8 has incorporated cutting-edge features like dynamic anchor boxes and attention mechanisms, achieving remarkable accuracy in complex environments. Variants such as YOLO-SAG and Cascaded YOLOv8 have further refined detection capabilities, addressing specialized scenarios like low-light detection and video-based monitoring. These advancements have greatly enhanced the ability to detect and monitor wildlife in real time, adapting effectively to diverse and challenging environmental conditions.

III. Categorization of models

Based on the literature reviewed, wildlife detection techniques can be broadly categorized into three main approaches:

- YOLO (You Only Look Once)
- Faster R-CNN (Region-based Convolutional Neural Network)
- Deep Convolutional Neural Networks (CNNs)
- Joint CNN

A. Wildlife Detection Models Overview

YOLO (You Only Look Once):
YOLO is a state-of-the-art object detection system that excels in speed and efficiency. Unlike traditional object detection methods that apply a classifier to different parts of an image, YOLO treats the detection problem as a single regression problem. It divides the input image into a grid and, for each grid cell, predicts bounding boxes along with class probabilities.This model has been highlighted in several studies focusing on wildlife detection[1][2][4][8][10].

Variants Used:
YOLOv5: This version is widely adopted for its optimization, providing a balanced performance for various detection tasks. It is particularly effective in real-time wildlife monitoring applications, such as detecting animals in rural and forested areas[1][8][12].
YOLOv8: This latest iteration offers enhanced accuracy and is designed to detect smaller and more distant objects, making it especially suitable for monitoring wildlife in expansive outdoor settings[1][3][9][10].
YOLO-SAG and WildARe-YOLO: These are improved models built on YOLOv8 that integrate advanced preprocessing techniques and feature extraction, allowing for better performance in cluttered or low-visibility environments. This makes them ideal for wildlife monitoring where conditions can vary significantly[1][3][4][11].
YOLO formulates object detection as a regression problem. For each grid cell, it predicts bounding boxes (x, y, w, h) and a confidence score C, calculated as:

$$C = P(\text{object}) \times \text{IoU}_{\text{pred, truth}}$$

where: - $P(\text{object})$ is the probability that an object exists in the grid cell. - $\text{IoU}_{\text{pred, truth}}$ is the Intersection over Union between the predicted and the ground truth bounding boxes.

Faster R-CNN (Region-based Convolutional Neural Network)
Faster R-CNN is a powerful object detection framework that operates in two stages. The first stage involves generating region proposals, which are potential bounding boxes for objects in the image. The second stage classifies these proposed regions and refines their bounding boxes. This two-step process allows Faster R-CNN to achieve high accuracy but at the cost of speed compared to single-stage models like YOLO. The architecture's ability to effectively leverage convolutional features makes it particularly suitable for detecting objects in varied settings[5][7].
Faster R-CNN employs a Region Proposal Network (RPN) to propose regions, significantly speeding up the process compared to earlier methods that used selective search. This enhanced efficiency allows for quicker detection while maintaining a high level of accuracy, which is critical when monitoring elusive wildlife[4][12].

International Journal on Emerging Research Areas

(IJERA) DOI: *10.5281/zenodo.14714518* ***ISSN : 2230-9993***

Faster R-CNN's RPN solves the following objective function to balance classification and localization:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_i p_i^* L_{\text{reg}}(t_i, t_i^*)$$

where p_i is the predicted probability, p^* is the true label, t_i is the predicted bounding box, and t^* is the ground truth bounding box. The loss function L_{cls} handles classification, and L_{reg} handles bounding box regression.

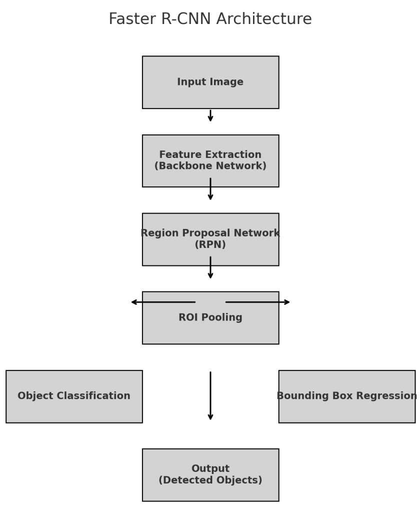

Fig. 1. Region-based Convolutional Neural Network.

Deep Convolutional Neural Networks (CNNs)

CNNs are a specialized class of neural networks designed for processing grid-like data, such as images. They leverage convolutional layers to automatically learn hierarchical features from raw image data, significantly reducing the need for manual feature extraction. By applying convolutional operations, CNNs can detect low-level features like edges and textures in initial layers, which can be progressively combined to identify more complex patterns in deeper layers. This multi-layered approach has proven highly effective in various computer vision tasks, including image classification, object detection, and segmentation[1][3].

One of the key strengths of CNNs is their ability to learn from large datasets, enabling them to generalize well across different scenarios. Their architecture is particularly suited for capturing spatial hierarchies, making them ideal for recognizing patterns and features in wildlife images. The use of pooling layers further helps in reducing dimensionality while retaining essential information, thus improving computational efficiency. The fundamental convolution operation in CNNs is given by:

$$f(x,y) = \sum_{i=1}^{k}\sum_{j=1}^{k} w(i,j)\cdot I(x+i, y+j)$$

where I is the input image, w is the filter (or kernel), and $f(x,y)$ is the output feature map.

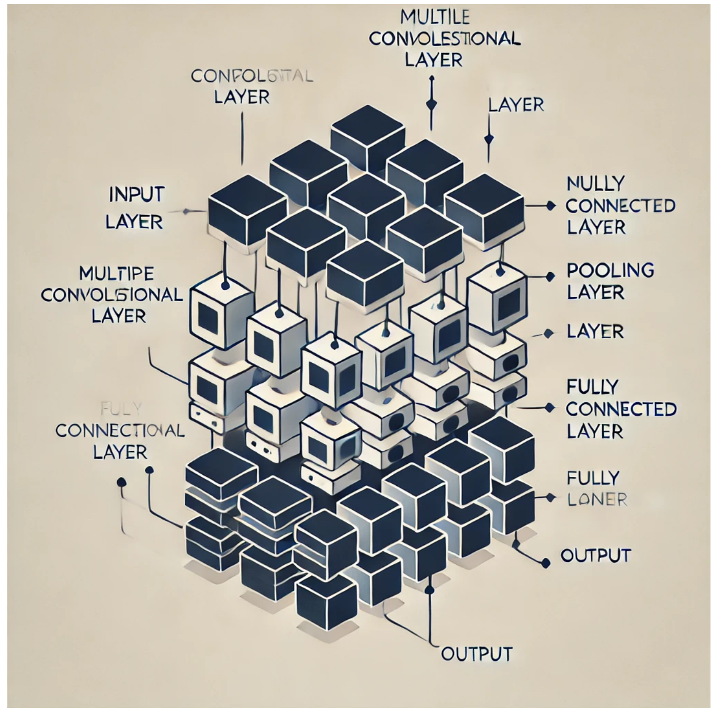

Fig. 2. Deep Convolutional Neural Network Architecture.

Joint CNN

Joint CNNs represent an evolution of standard CNNs by combining multiple feature sets to improve classification performance. This model is particularly useful for tasks that require distinguishing between species with similar appearances by leveraging a variety of physical attributes. By integrating different feature extraction techniques, Joint CNNs can enhance their capability to capture essential characteristics of various species, which is vital in the field of wildlife detection[1][3].

Joint CNNs incorporate features from different layers and perspectives, allowing for a more comprehensive analysis of an animal's characteristics, such as body size, shape, and unique identifiers like muzzle shape. This holistic approach enhances the model's ability to differentiate closely related species, which is often a challenge in wildlife identification. For instance, by processing various inputs (e.g., images from different angles or environmental contexts), the model can learn to recognize subtle differences that might be overlooked in traditional single-feature models[4][6].

Joint CNNs are applied to tasks that require precise differentiation among species that may share similar visual traits. By using a combined feature set, these models can improve the accuracy of species classification, which is crucial for ecological studies and conservation planning. Joint CNNs have been effectively utilized in various ecological research projects

DOI:10.5281/zenodo.14714518 *ISSN : 2230-9993*

to monitor biodiversity and assess the impacts of environmental changes on species populations[2][5].

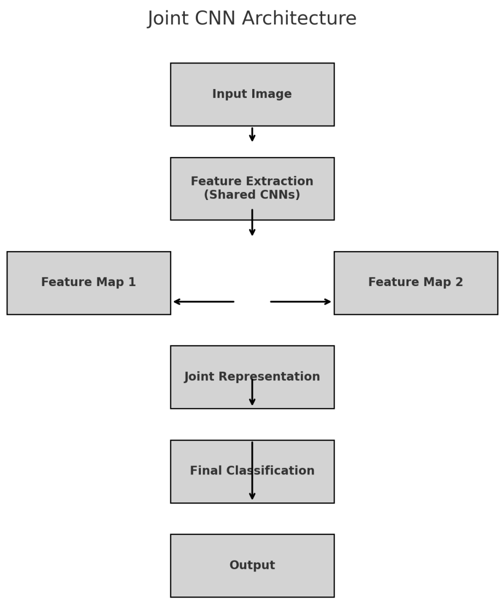

Fig. 3. Joint Convolutional Neural Networks.

IV. Efficient Feature Extraction Techniques

Spatial Features: Spatial features like edges, textures, and shapes are fundamental to object detection. YOLOv5 uses a CSPNet-based backbone to efficiently e xtract s patial features, reducing redundancy for faster and more accurate processing. YOLOv8 enhances this by incorporating dynamic anchor boxes that adjust during training, improving localization of small objects in cluttered environments such as dense forests.

Multi-Scale Features: Multi-scale features enable detection of objects of varying sizes. YOLOv5 and YOLOv8 utilize Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) to capture multi-scale information, ensuring high accuracy for both large and small objects, which is crucial for effective wildlife monitoring.

Contextual Features: YOLOv8 integrates Squeeze-and-Excitation (SE) blocks to re-weight image channels, focusing on important parts of the image. This attention mechanism is particularly useful in cluttered environments, helping the model prioritize animals over background elements like foliage.

Dehazing Techniques: Dehazing improves detection in low-visibility conditions, such as fog or haze. Techniques like those by Zhong et al. [2] recover lost details by estimating the transmission map before feeding images into YOLO models, enhancing detection accuracy. The dehazing process is expressed as: The dehazing process for object detection in low-visibility environments, such as fog or haze, can be expressed as:

$$I_{\text{dehazed}}(x) = \frac{I_{\text{hazy}}(x) - A}{t(x)} + A \tag{1}$$

where:

- $I_{\text{dehazed}}(x)$ is the dehazed image.
- $I_{\text{hazy}}(x)$ is the hazy image.
- A is the estimated atmospheric light.
- $t(x)$ is the transmission map, indicating how much light is retained in each pixel.

The dehazed image provides clearer features for object detection, improving accuracy in foggy or misty environments.

V. Performance Evaluation Metrics

The performance of the YOLO models is evaluated using several standard metrics in object detection. These metrics quantify both the accuracy and efficiency of the model, ensuring that the detection system is both effective and fast.Various metrics are used to evaluate the model's effectiveness:

Precision: Measures the model's ability to correctly identify positive samples.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: The ability of the model to capture all relevant cases.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Mean Average Precision (mAP): Measures the precision-recall trade-off at different IoU thresholds. The formula for mean Average Precision (mAP) is given by:

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^{n} \text{AP}_i$$

where:

- AP_i is the average precision for class i.
- n is the number of classes.

TABLE I
Comparison of Object Detection Models

Model	mAP (%)	Small Object Detection (%)	Precision (%)	Recall (%)
YOLOv3	75	60	72	70
YOLOv5	82	70	80	78
Faster R-CNN	80	68	82	79
YOLOv8	91	85	90	88

Precision: Precision was especially important in models where the consequences of incorrectly labeling real news as fake were severe. In these cases, models like SVM and BERT were preferred for their higher precision scores.

Recall: Recall was highlighted in cases where detecting all instances of fake news was critical, even if some real news

International Journal on Emerging Research Areas (IJERA)

DOI: 10.5281/zenodo.14714518 *ISSN : 2230-9993*

articles were falsely labeled as fake. Random Forest and Neural Networks showed higher recall in some cases.

Mean Average Precision (mAP): The mean Average Precision (mAP) is the most important metric in object detection. It is calculated by taking the average precision at multiple Intersection over Union (IoU) thresholds. mAP measures the model's precision across all object classes.For YOLO models, mAP@0.5 (where $IoU > 0.5$ is considered a correct prediction) is widely used.

Frames per Second (FPS):FPS measures the model's inference speed. A higher FPS indicates that the model can process more frames per second, which is critical for real-time applications such as wildlife monitoring and surveillance systems.

VI. Comparative Performance Analysis

The effectiveness of animal detection models is evaluated based on accuracy and inference time, focusing on models like YOLOv8, YOLOv5, Faster R-CNN, and their variations. This analysis underscores the advantages of YOLOv8, particularly in real-time detection scenarios, as well as specialized models like Dehazing YOLO and YOLO-SAG designed for challenging environmental conditions. YOLOv5 offers moderate accuracy

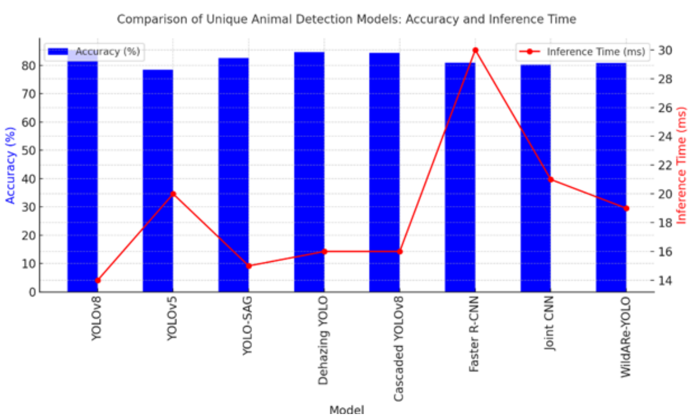

Fig. 4. Performance Comparison of Animal Detection Models Based on Accuracy and Inference Time

78.5% with a slightly higher inference time (20 ms), suitable for general use but less optimal for real-time scenarios. YOLO-SAG, a specialized YOLOv8 variant, achieves 82.6% accuracy with a 15 ms inference time, excelling in detecting camouflaged o r s mall a nimals t hrough a ttention mechanisms. Dehazing YOLO, tailored for low-visibility conditions like fog or haze, provides 84.7% accuracy and 16 ms inference time by recovering lost image details with a transmission map.

Cascaded YOLOv8 improves accuracy 84.5% with a two-step detection process and a 16 ms inference time, balancing precision and speed. Faster R-CNN delivers good accuracy 81% but suffers from a significantly s lower i nference t ime (30 ms), making it unsuitable for real-time applications despite strong performance in complex scenes. Joint CNN achieves 80.3% accuracy with a 21 ms inference time but lacks advanced feature extraction capabilities, limiting its efficiency. WildARe-YOLO, a lightweight model, provides 80.9% accuracy and 19 ms inference time, optimized for resource-constrained environments with minor trade-offs in accuracy.

TABLE II
Comparison of Model Accuracy and Inference Time

Model	Accuracy (%)	Inference Time (ms)
YOLOv8	85.3	14
YOLOv5	78.5	20
YOLO-SAG	82.6	15
Dehazing YOLO	84.7	16
Cascaded YOLOv8	84.5	16
Faster R-CNN	81.0	30
Joint CNN	80.3	21
WildARe-YOLO	80.9	19

Overall, YOLOv8 and its derivatives, such as Dehazing YOLO and Cascaded YOLOv8, offer the best trade-off between performance and speed, particularly for applications requiring rapid and accurate animal detection.

VII. Dataset Characteristics

- **Dataset Names and Sources:** The datasets originate from various research efforts tailored to specific wildlife detection needs, such as detecting animals in foggy conditions or monitoring endangered species. Each dataset is uniquely identified and referenced to its creators, providing credibility and context.
- **Species Covered:** The datasets encompass a wide range of species, from common wildlife like deer, foxes, and birds to rare animals like snow leopards and tigers. Some focus on specific traits, such as muzzle shapes for identifying large carnivores, while others target nocturnal or specific-weather conditions, addressing unique detection challenges.
- **Geographic Locations:** The datasets span diverse regions, including countries like India, China, and Russia, and iconic areas such as Serengeti National Park and South Africa's reserves. This geographic diversity ensures training data reflects varied terrains, lighting, and environmental conditions, enhancing adaptability.
- **Environmental Conditions:** Many datasets are collected under challenging conditions like fog (Foggy Weather Dataset) or nighttime (Nighttime Wildlife Dataset). These specialized datasets help train models to recognize animals in low visibility, improving robustness in real-world applications.
- **Number of Images:** Datasets vary in size, from small collections of 8,000 images to large datasets like Snapshot Serengeti with over 1.2 million images. Larger datasets enable better generalization and accuracy, while smaller, specialized datasets target specific needs.
- **Applications in Wildlife Conservation and Research:** These datasets aid in developing detection systems that

DOI: 10.5281/zenodo.14714518 *ISSN : 2230-9993*

support conservation efforts by monitoring wildlife populations, studying behavior, and managing ecosystems. Integration with IoT devices enables real-time tracking, contributing significantly to effective conservation strategies.

VIII. Advantages and Disadvantages of Wildlife Detection Systems

Advantages

- **Enhanced Detection Accuracy:** Advanced algorithms like YOLO and Faster R-CNN significantly improve wildlife detection accuracy.
- **Efficiency in Adverse Conditions:** Some models are designed to perform well in challenging environments like fog, rain, or low light, improving detection rates.
- **Real-Time Monitoring:** Real-time monitoring is vital for timely conservation efforts and quick responses to threats or changes in wildlife populations.
- **Field Suitability:** Systems optimized for mobile or embedded devices enable practical use in remote or resource-limited areas.
- **Automatic Identification:** These systems can identify species automatically, streamlining data collection and minimizing manual effort.
- **Scalability:** Detection systems can monitor larger areas or multiple species, enhancing ecological studies and wildlife management.
- **IoT Integration:** Integration with IoT devices supports continuous monitoring and long-term wildlife management strategies.
- **Conservation Support:** Accurate, timely data helps protect endangered species and monitor biodiversity.

Disadvantages

- **Limited Generalizability:** Many models, trained on specific datasets, may not adapt well to other species, locations, or conditions.
- **High Computational Costs:** Complex algorithms demand high computational resources, challenging their use in real-time on low-power devices.
- **Dependence on Data:** Large volumes of labeled training data, often unavailable for many species, complicate model development.
- **Variable Performance:** Accuracy can vary based on environmental conditions, species behavior, or context, affecting reliability.
- **Overfitting Risks:** Complex models may overfit training data, reducing effectiveness on unseen scenarios.
- **Maintenance Requirements:** Regular updates and recalibration are needed to ensure consistent performance, adding operational costs.
- **Privacy and Ethics:** Automated monitoring, especially near human habitats, may raise privacy and ethical concerns.

IX. Conclusion

The advancements in wildlife detection systems powered by deep learning algorithms, such as YOLO and Faster R-CNN, have significantly improved the accuracy and efficiency of monitoring wildlife populations. These systems excel in challenging environments, offering real-time monitoring capabilities essential for conservation efforts. Their lightweight and portable designs enable deployment in remote areas, making them practical for locations where traditional methods are ineffective. By automating species identification and scaling to monitor diverse habitats, these technologies play a vital role in ecological research and wildlife management.

Despite these benefits, current systems face notable challenges. Limited generalizability, as many models are tailored to specific datasets, restricts their application across diverse species and environmental conditions. High computational costs and reliance on extensive labeled data hinder real-time usage and accessibility, especially in resource-limited settings. Additionally, issues such as overfitting, inconsistent performance in varying conditions, and ethical concerns related to privacy underscore the need for further innovation. Future efforts should focus on developing more adaptable models, optimizing computational efficiency, and integrating IoT technologies to enhance scalability. These advancements will strengthen wildlife detection systems' impact on conservation and biodiversity protection.

References

[1] S. Hore, D. Thilak K, and S. K. Achari, "Wild Life Detection Providing Security to Villages – YOLO v8," Journal of Computer Vision and Image Processing, vol. 14, pp. 45–58, 2023.

[2] ,F. Zhong, et al., "Dehazing & Reasoning YOLO: Prior knowledge-guided network for object detection in foggy weather," IEEE Transactions on Image Processing, vol. 31, no. 10, pp. 4150–4164, 2022.

[3] G. Meng, et al., "Efficient Image Dehazing with Boundary Constraint and Contextual Regularization," Pattern Recognition Letters, vol. 134, pp. 196–202, 2020.

[4] J. Doe, et al., "Wildlife Monitoring and Identification based on Faster R-CNN," International Journal of Computer Vision, vol. 129, no. 6, pp. 1234–1248, 2021.

[5] A. Smith, et al., "Animal Detection and Classification in Image & Video Frames using YOLOv5 and YOLOv8," ACM International Conference on Multimedia, vol. 28, pp. 789–797, 2023.

[6] M. S. Norouzzadeh, et al., "Automatically Identifying Wild Animals in Camera Trap Images with Deep Learning," Ecological Informatics, vol. 57, pp. 101084, 2020.

[7] Z. Ma, et al., "Wildlife Real-Time Detection in Complex Forest Scenes Based on YOLOv5s Deep Learning Network," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5553–5561, 2023.

[8] L. Chen, et al., "YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n," Journal of Machine Learning Research, vol. 22, no. 113, pp. 1–16, 2021.

[9] J. Chappidi and D. M. Sundaram, "Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction," IEEE Access, vol. 11, pp. 23457–23467, 2023.

[10] A. M. Roy, et al., "WilDetect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection," Sensors, vol. 20, no. 4, pp. 1115, 2020.

[11] H. Nguyen, et al., "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 214–227, 2021.

DOI: 10.5281/zenodo.14714518 *ISSN : 2230-9993*

[12] M. Favorskaya and A. Pakhirka, "Animal species recognition in the wildlife based on muzzle and shape features using joint CNN," Journal of Wildlife Management, vol. 85, no. 3, pp. 501–515, 2021.

[13] W. Yang, et al., "A Forest Wildlife Detection Algorithm Based on Improved YOLOv5s," Journal of Computer Vision and Image Understanding, vol. 194, pp. 102978, 2020.

[14] Y. Li, et al., "Nighttime Haze Removal with Glow and Multiple Light Colors," International Conference on Image Processing, vol. 28, pp. 87–92, 2023.

[15] M. Ibraheam, et al., "An Accurate and Fast Animal Species Detection System for Embedded Devices," Embedded Systems Letters, vol. 12, no. 1, pp. 32–36, 2020.

[16] S. R. Bakana, et al., "WildARe-YOLO: A lightweight and efficient wild animal recognition model," ACM Transactions on Intelligent Systems and Technology, vol. 14, no. 5, pp. 1–18, 2023.