Smart Wildlife Detection Framework Using YOLOv8 Segmentation for Conflict Mitigation

[1]Dr. K. Prabavathy, [2]Mrs. D. Subha Shree
[1]Assistant Professor & Head, Department of B.Sc. (Data Science and Analytics) Sree Saraswathi Thyagaraja College Pollachi
[2]Research Scholar, Sree Saraswathi Thyagaraja College Pollachi

***Abstract*—Human–wildlife interactions have intensified in many rural and forest-bordering communities due to widespread habitat destruction and ongoing deforestation [1,2,3,4,5]. These encounters frequently cause crop losses, property damage, and serious risks to human lives. Conventional surveillance mechanisms often lack the precision and response speed needed to identify wildlife intrusions effectively, resulting in delayed interventions and increased danger for both people and animals. To address this gap, this work introduces an enhanced YOLOv8 segmentation framework integrated with a CSPDarkNet53 backbone. By partitioning feature maps and minimizing redundant computation, CSPDarkNet53 strengthens feature extraction while preserving essential information. The model employs multi-scale convolutional filters (3×3 and 5×5), batch normalization, and Leaky ReLU for stable learning, along with a PANet-based neck that improves multi-level feature aggregation. Experimental analysis demonstrates superior segmentation and detection performance, achieving a bounding box mAP of 0.86 and mask mAP of 0.79—outperforming YOLOv8-M and YOLOv10-M. High recall values for elephants (0.841) and tigers (0.891) further validate the model's robustness. While segmentation for elephants is consistently accurate, occasional tiger–background misclassifications highlight areas for improvement. Overall, the proposed model offers a practical and efficient approach to wildlife monitoring and conflict mitigation.**

***Index Terms*—Human-Animal conflicts, YOLO, Segmentation, Darknet53, PANet and Proposed YOLOv8 Segmentation**

I. INTRODUCTION

Conflicts between humans and wildlife have become a major concern in regions adjacent to forests, where animals regularly wander into farmlands and villages, damaging crops and property and

posing threats to human safety animal segmentation. These incidents have escalated due to shrinking natural habitats and increasing human encroachment. Monitoring such interactions manually or through basic motion-detection systems is often unreliable since these systems struggle to accurately detect animal movement or provide timely alerts. Consequently, delays in detection raise the likelihood of accidents, injuries, and harm to wildlife.

To overcome these limitations, this study proposes a YOLOv8-based segmentation model strengthened by a CSPDarkNet53 backbone. CSPDarkNet53 enhances feature extraction by splitting feature maps, thereby reducing unnecessary computations while retaining critical details. The architecture utilizes multiple convolutional kernels to capture objects of varying scales and employs batch normalization and Leaky ReLU for stable training dynamics. The integration of a PANet neck further improves multi-level feature fusion, enabling efficient detection of both large and small animals [7][8].

Initial evaluations show that the proposed model significantly enhances detection accuracy, achieving high bounding box and mask mAP scores. Precision–recall behaviour indicates strong performance in detecting elephants and tigers, although some tiger instances are occasionally confused with background features. Despite these challenges, the model demonstrates considerable potential for use in real-time conflict-prevention systems [1],[2],[3],[5],[6],[10].

II. RELATED WORKS:

Several studies have explored deep-learning-based wildlife monitoring to reduce human–animal conflicts. One work applies YOLOv8 to detect wildlife intrusions in agricultural zones, enabling the system to alert farmers instantly and thereby reduce crop losses and animal harm [11], [12]. Another system integrates YOLOv5 with auditory deterrents—such as simulated gunfire—to discourage wildlife from entering restricted zones [12].

Other research focuses on improving segmentation accuracy. For example, a deep CNN with genetic segmentation achieved high precision and recall for animal detection using a small handcrafted dataset [13]. Marine-life segmentation studies also report improved performance using Siamese networks, advanced augmentations, and models like MAS-SAM, which enhance the Segment Anything Model for underwater environments [14], [15], [19], [21].

Further advancements include I-MedSAM, which combines implicit neural representations with SAM for medical segmentation [17], [22], [23], and synthetic-animal datasets that expand training diversity for part-based segmentation tasks [18]. In fisheries research, coordinate-aware Mask R-CNN variants demonstrate improvements in detecting underwater species using specialized normalization and loss functions [19]. Recent YOLO-based models also emphasize improved lightweight architectures—such as optimized YOLOv5 variants—to support high-speed detection useful for real-time wildlife management [6], [10], [20].

III. DATASET FOR PROPOSED YOLOV8 SEGMENTATION

• Dataset Collection:

The dataset for this study was created from publicly available YouTube videos capturing tiger–elephant interactions animal segmentation. Frames were extracted at a rate of one per ten frames to balance dataset size with content diversity. All images were resized to 640×640 pixels to ensure uniformity during training. This approach allows the dataset to capture wide-ranging environmental conditions, improving model generalization. Figure 1 in the original document illustrates example frames and their segmented outputs [11].

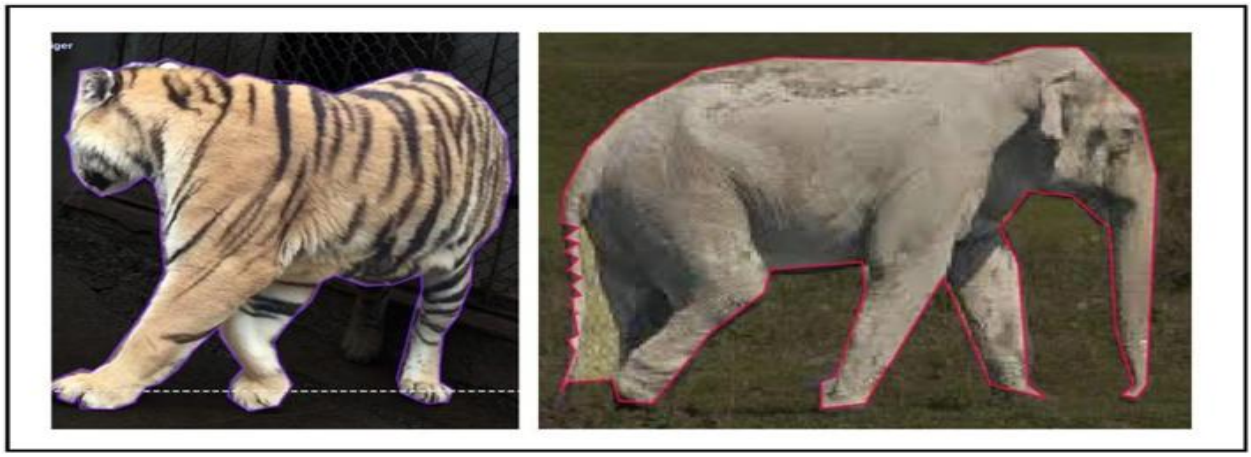

Fig 1 Sample Collected images from Open source for animal segmentation

• Data PreProcessing:

To improve robustness, the dataset underwent extensive augmentation, including horizontal flips, up to 20% zoom adjustments, ±10° shearing in both axes, hue shifts within ±15°, and saturation modifications up to ±25%. These augmentations help the model adapt to varied lighting, perspective changes, and animal poses.

After preprocessing, the data was divided into training (70%), validation (20%), and testing (10%) sets, ensuring a balanced evaluation strategy. Table 1 outlines the final distribution of tiger and elephant images across the splits [6][7].

Table 1 Dataset splitting ratio for proposed segmentation model

Dataset splitting	Tiger	Elephant
Training	954	954
Validation	180	184
Testing	50	46

IV. PROPOSED YOLOV8 SEGMENTATION MODEL:

The proposed architecture maintains the standard YOLOv8 structure—Backbone, Neck, and Head—while incorporating modifications for improved performance in wildlife segmentation tasks.

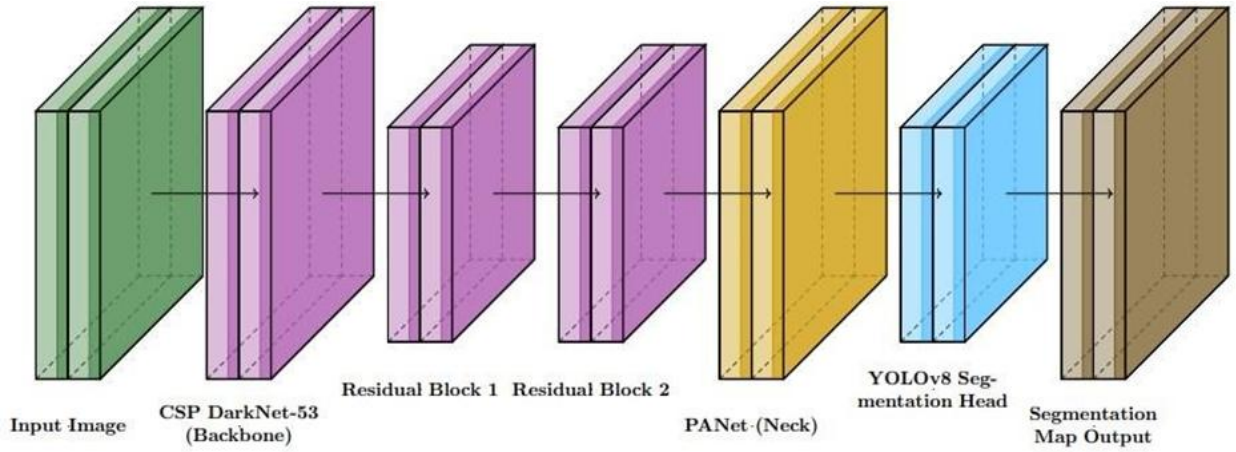

Fig 2 Proposed YOLOv8 Segmentation Model

Figure 2 shows the proposed YOLOv8 segmentation model, which follows the traditional structure with three main parts: Backbone, Neck, and Head. The Backbone extracts important features from the input image, the Neck enhances these features using methods like Feature Pyramid Networks (FPN) to detect objects at different sizes, and the Head makes final predictions, including object categories, bounding boxes, and segmentation masks. In this improved model, the Backbone is modified to increase accuracy and capture better details. Figure 3 highlights how this upgraded YOLOv8 segmentation model is designed for animal monitoring, including a system for tracking both humans and animals [6][7][10][20].

Backbone - CSP DarkNet 53:

CSPDarkNet53 is utilized for its efficiency in deep feature extraction. It begins with convolutional operations that transform the input image into foundational feature maps. Residual connections help maintain gradient flow and reduce information loss during training. The CSPNet mechanism divides feature maps into two streams, processing them separately and recombining them to reduce redundant computations. Multi-scale kernels capture finer and broader details, batch normalization stabilizes learning, and Leaky ReLU introduces controlled non-linearity.

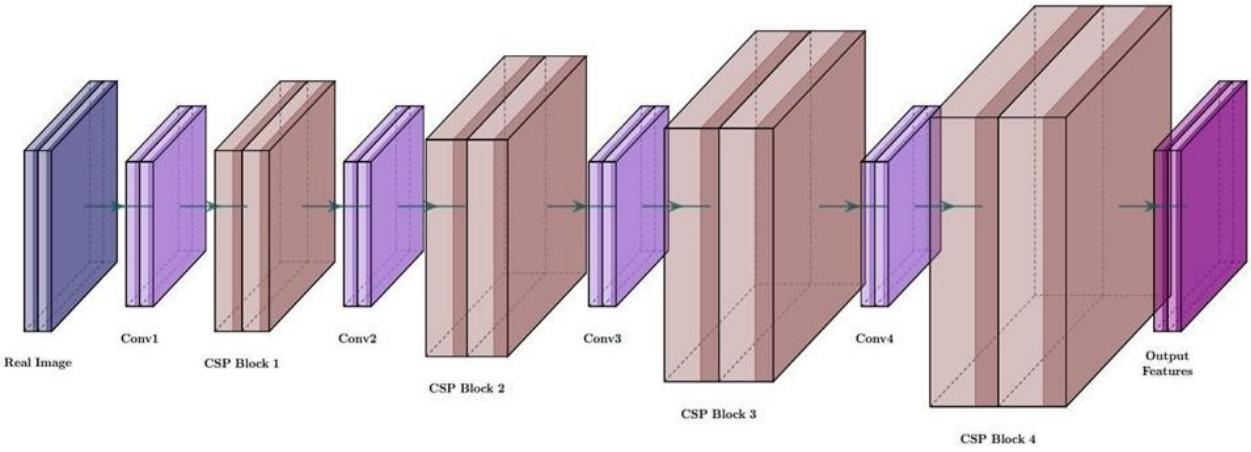

Fig 3 CSPDarkNet53 Architecture Backbone for Proposed YOLOv8 Segmentation

Fig 3 showcase proposed YOLOv8 segmentation network begins with an initial convolutional layer that processes the raw input image, extracting basic features such as edges and textures. This layer applies multiple filters to the image to produce a feature map. Mathematically, this operation can be represented in equation 1

$$S_{out1} = \mathbf{Conv}(S_{input}, \mathbf{W}, b) \quad 1$$

output S^{input} is a feature map generated from convolution from the filters. The network depends where denotes the input image, W is the convolutional weights, and *b* is the bias term. The on residualS^{out1} blocks to maintain the forward propagation of pertinent information as it becomes deeper. Block Residual The residual block adds the input feature map back to the convolutional layers' output in order to avoid the vanishing gradient issue. This operation is represented mathematically as in equation 2

$$S_{residual} = S_{input} + \mathbf{Conv}(S_{input}) \quad 2$$

This addition allows the model to retain essential properties while improving the learning process. One of the key breakthroughs of CSPDarkNet53 is the use of Cross-step Partial Networks (CSPNet), which divide the feature map at each step. One component continues through the convolutional layers, while the other part is subjected to residual processing. The two pieces are then blended back together, significantly lowering computational complexity while maintaining useful features. The split and merge procedure are mathematically represented as equation 3, In the equation 3 represents a merge operation in layer, it comes derived from a splitted operations in layer

$Ssplit1, Ssplit2$ = Split ($Sinput$), $Sout1$ = Conv ($Ssplit1$), $Sout2$ = Conv ($Ssplit\ 2$), $Smerged$ = Concat ($Sout1, Sout2$) 3

This effectively avoids redundant operations, preserves features that are necessary, and quickens the pace while improving the accuracy. It also extracts multi-scale features of different kernel sizes in convolutional layers (3x3 and 5x5), considering that objects can be of any size. Batch normalization normalizes the output of each convolution to have the same distribution of features. A batch normalization operation could be written as in equation 4:

$$x_{norm} = \frac{x - \mu}{\sigma} \cdot \gamma + [\text{Tab}] \quad 4$$

Where equation 4 as μ and σ are the mean and standard deviation of the batch, while γ and β are learnable scaling and shifting parameters. Finally, the activation function Leaky ReLU is applied to introduce non-linearity. In CSP Darknet 53 connects with a PaNet (Neck) part of the proposed YOLOv8 segmentation model. In Backbone will extract the in-depth details along with feature extraction and processing in a given dataset. Backbone has splitted into three parts of connections to the Neck part of YOLOv8 by its input resolution size of small, medium and large connection will extract object detection with segmentation.

The flow of connection backbone to Path Aggregation Network (PANet) in YOLOv8's Neck is

designed to improve feature fusion by allowing information to flow in two directions: top-down and bottom-up. The top-down path works by upsampling high-level features (which contain semantic information) and combining them with low-level features (which have more spatial details). This helps YOLOv8 capture both the broad context of an image and the fine details. The bottom-up path helps by taking spatial information from lower layers and adding it back to the higher layers, making the model better at detecting smaller objects with high precision. This bi-directional flow improves multi-scale feature detection, which is crucial for accurately identifying objects of different sizes.

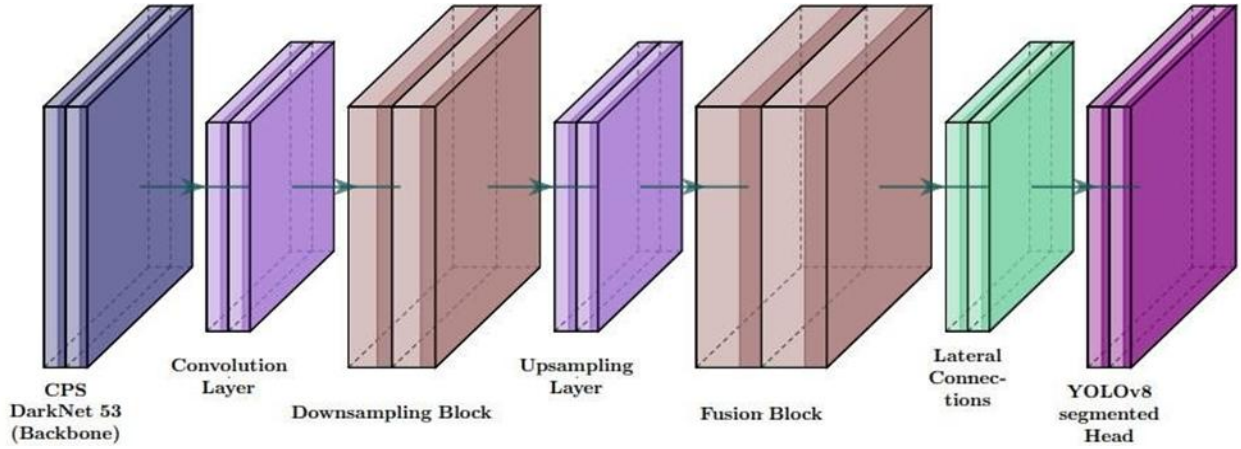

Fig 4 PANet (Neck part of proposed YOLOv8 Segmentation)

Fig 4 represent an PANet top - down upsampling high level feature extraction process showcase in mathematical process represent in equation 5

Si = Concat (Downsample($Si-1$), Si) 5

By adding this mechanism, PANet helps YOLOv8 focus on important features from both high and low levels, improving object detection accuracy. This makes it easier for the model to handle complex scenarios with objects of varying sizes and orientations, making it more efficient and less computation in processing. PANet has passed three connections with Head for object detection and segmentation process for various sizes for instance segmentation.

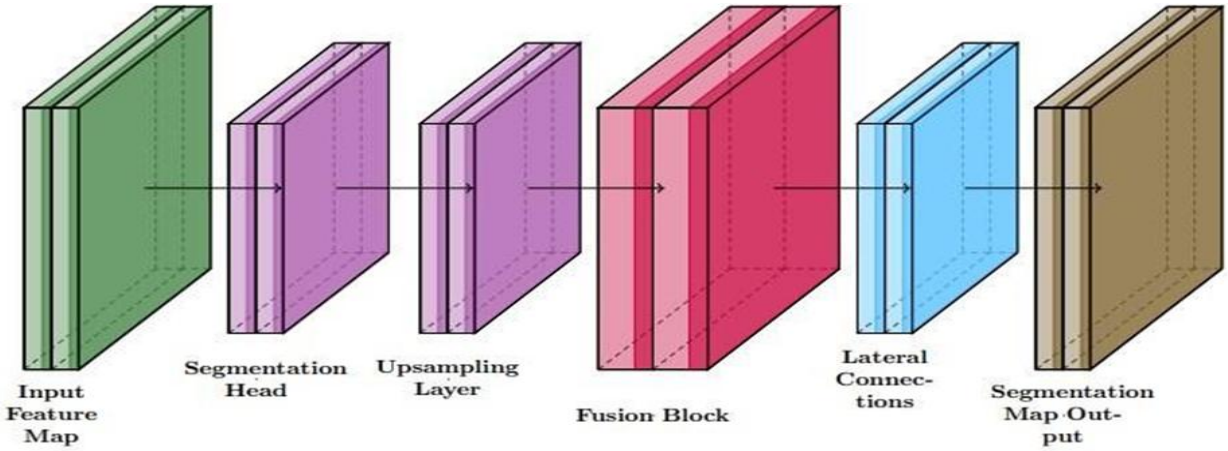

Fig 5 YOLOv8 Segmentation Head

In fig 5 showcase a YOLO Head part handles both object detection and segmentation tasks by predicting key components from the feature maps processed by the Backbone and Neck. It generates predictions for class labels *C*, Objectness scores $Pobj$, Bounding Box (BB) coordinates (x,y,w,h), and segmentation makes *S*. The prediction output for each detected object can been expressed in equation 6:

Output = $\{(xi, yi, wi, hi, Pobj, Ci, Si)\}$ - 6

In equation 6 represents a proposed YOLOv8 segmentation output for object detection and segmentation result for animal monitoring to prevent human animal conflicts. In comes to detection or head part of proposed YOLOv8 divided by two parts of outcomes are Object detection and segmentation. In equation (xi, yi, wi, hi) are bounding box coordinates for detection, $Pobj$ is Objectness score, Ci is class label and Si is segmentation mask for the i-th object in instance segmentation output of proposed YOLOv8 Model. The Segmentation Head combines convolutional layers and up sampling to refine these predictions and generate precise segmentation masks that outline the detected objects with clear boundaries. This structure enables YOLOv8 to provide both accurate object detection and segmentation in real-time, making it highly efficient for tasks requiring both detection and precise delineation, such as autonomous driving, medical imaging, and surveillance.

V. EXPERIMENTAL RESULT AND DISCUSSION

- Performance matrices:

The enhanced YOLOv8 segmentation model shows strong performance across both detection and segmentation tasks. Training visualizations (Fig. 6) indicate stable convergence. The F1-Confidence curve (B) achieves 0.84, precision reaches 1.00 at a high threshold, and recall peaks at 0.89 for all classes. Precision–recall analysis indicates reliable detection of elephants and tigers, with an overall bounding box mAP of 0.866[11][12].

Segmentation results show a mask F1 score of 0.79 at a confidence threshold of 0.571, and mask precision reaches 1.00 at 0.967 confidence. The model achieves class-wise mask mAPs of 0.723 for elephants and 0.851 for tigers, with an overall mAP of 0.787.

The confusion matrix indicates excellent elephant detection but moderate confusion between tigers and background elements, suggesting the need for improved feature discrimination.

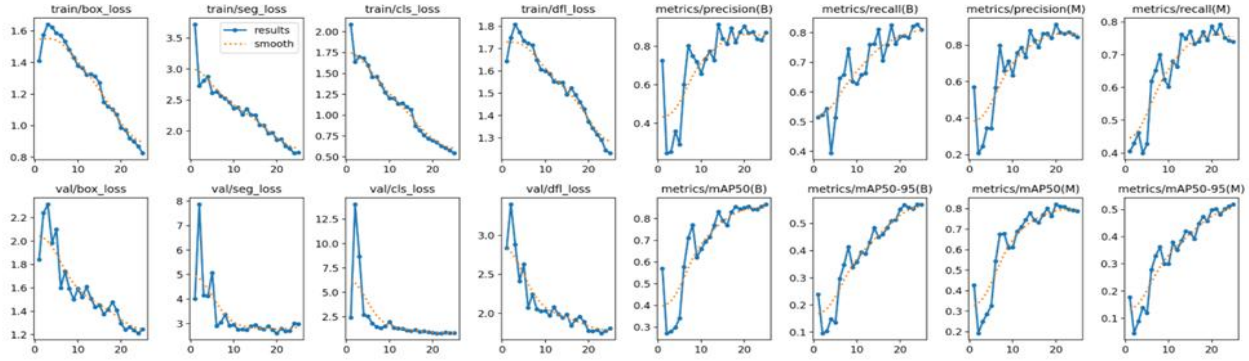

Fig 6 Result images of proposed YOLOv8 Segmentation

Figure 6 presents result images that graphically represent the training and validation performance of the proposed YOLOv8 segmentation algorithm. This approach combines object detection and segmentation. For instance, segmentation, it tracks training and validation losses, including box_loss for object detection and seg_loss for mask segmentation. The result graphs also include key object detection metrics like precision (B), recall (B), and mAP50 (B), while segmentation metrics include precision (M), recall (M), mAP50 (M), and mAP50-95 (M).

Table 2 Animal Segmentation model performance comparison

Model	Bounding Box (mAP - %)	Mask (mAP - %)
Proposed YOLOv8 Segmentation Model	0.86	0.79
YOLOv8 - M	0.82	0.73
YOLOv10 - M	0.85	0.77

As shown in Table 2, the proposed YOLOv8 segmentation model outperforms YOLOv8-M and YOLOv10-M in both bounding box and mask mAP. The improved backbone and feature fusion strategies contribute to this enhanced accuracy.

VI. CONCLUSION

The integration of CSPDarkNet53 and PANet within the YOLOv8 segmentation framework presents a highly effective solution for wildlife detection and monitoring. The model achieves strong detection and segmentation metrics, demonstrating superior performance over comparable YOLO variants. While some tiger–background confusion persists, the overall results show that this model is well-suited for real-time human–wildlife conflict prevention. Its ability to detect animals accurately can significantly improve safety for both humans and wildlife in vulnerable regions.

REFERENCES

[1] R. K. Pandey, S. P. Yadav, K. M. Selvan, L. Natarajan, and P. Nigam, “Elephant conservation in India: Striking a balance between coexistence and conflicts,” *Integrative Conservation,* vol. 3, no. 1, pp. 1–11, 2024.

[2] B. Bhagabati, K. K. Sarma, and K. C. Bora, “An automated approach for human-animal conflict minimisation in Assam and protection of wildlife around the Kaziranga National Park using YOLO and SENet Attention Framework,” Ecological Informatics, vol. 79, p. 102398, 2024.

[3] R. Ramakrishnan, S. Rajendrakumar, and N. K. Kothurkar, “Regional sustainability of the Kattunayakan tribe in Kerala, India through the enhancement of agricultural, livestock, and livelihood options,” Agricultural Systems, vol. 217, p. 103929, 2024.

[4] L. G. Jayaprakash and G. M. Hickey, “Elephants in the Room—Analyzing Local Discourses for Sustainable Management of Bannerghatta National Park, South India,” Environmental Management, pp. 1–21, 2024.

[5] A. Bhengra and A. Bhengra, "Understanding the Human Toll of Human-Elephant Conflicts: Insights from a Four-Year Autopsy Analysis," Cureus, vol. 17, no. 1, 2025.

[6] X. Guo, F. Jiang, Q. Chen, Y. Wang, K. Sha, and J. Chen, "Deep learning-enhanced environment perception for autonomous driving: MDNet with CSP-DarkNet53," Pattern Recognition, vol. 160, p. 111174, 2025.

[7] Y. Pan, G. Wang, and J. Yu, "Overview of deep learning YOLO algorithm," in Proc. 4th Int. Conf. on Computer Vision, Application, and Algorithm (CVAA 2024), SPIE, vol. 13486, pp. 622–630, Jan. 2025.

[8] P. T., R. Thangaraj, P. P., U. R. M., and B. Vadivelu, "Real-Time Handgun Detection in Surveillance Videos Based on Deep Learning Approach," in Proc. 2022 Int. Conf. on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 689–693, doi: 10.1109/ICAAIC53929.2022.9793288.

[9] F. Li, T. Sun, Q. Liu, H. Si, G. Zheng, Q. Wang, et al., "A New Backbone Network for Improving Faster R-CNN Detection Accuracy," SSRN Electronic Journal, 2023, doi: 10.2139/ssrn.4536358.

[10] L. Deng, H. Li, H. Liu, and J. Gu, "A lightweight YOLOv3 algorithm used for safety helmet detection," Scientific Reports, vol. 12, no. 1, p. 10981, 2022.

[11] K. M., B. B., and A. K., "Animal Intrusion Detection Using YOLO V8," in Proc. 2024 10th Int. Conf. on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 206–211, 2024, doi: 10.1109/ICACCS60874.2024.10716895.

[12] M. Kumar, M. Aslam, and M. Akshay, "Advanced Wild Animal Detection and Alert System Using YOLO V5 Model," in Proc. 2023 7th Int. Conf. on Trends in Electronics and Informatics (ICOEI), pp. 365–371, 2023, doi: 10.1109/ICOEI56765.2023.10126065.

[13] R. Chandrakar, R. Raja, and R. Miri, "Animal detection based on deep convolutional neural networks with genetic segmentation," Multimedia Tools and Applications, pp. 1–14, 2022.

[14] Z. Fu, R. Chen, Y. Huang, E. Cheng, X. Ding, and K. Xu, "MASNet: A Robust Deep Marine Animal Segmentation Network," IEEE Journal of Oceanic Engineering, vol. 49, pp. 1104–1115, 2024, doi: 10.1109/JOE.2023.3252760.

[15] T. Yan, Z. Wan, X. Deng, P. Zhang, Y. Liu, and H. Lu, "MAS-SAM: Segment Any Marine Animal with Aggregated Features," 2024, doi: 10.48550/arXiv.2404.15700.

[16] X. Wei, J. Cao, Y. Jin, M. Lu, G. Wang, and S. Zhang, "I-MedSAM: Implicit Medical Image Segmentation with Segment Anything," arXiv preprint arXiv:2311.17081, 2023, doi: 10.48550/arXiv.2311.17081.

[17] J. Tang, Y. Zhao, L. Feng, and W. Zhao, "Contour-based wild animal instance segmentation using a few-shot detector," Animals, vol. 12, no. 15, p. 1980, 2022.

[18] L. X. Estévez-Moreno, G. C. Miranda-de la Lama, and G. G. Miguel-Pacheco, "Consumer attitudes towards farm animal welfare in Argentina, Chile, Colombia, Ecuador, Peru and Bolivia: A segmentation-based study," Meat Science, vol. 187, p. 108747, 2022.

[19] J. Peng, J. He, P. Kaushik, Z. Xiao, J. Mu, and A. Yuille, "Learning Part Segmentation from Synthetic Animals," in Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision

(WACV), pp. 90–101, 2024.

[20] D. Yi, H. B. Ahmedov, S. Jiang, Y. Li, S. J. Flinn, and P. G. Fernandes, "Coordinate-Aware Mask R-CNN with Group Normalization: An underwater marine animal instance segmentation framework," Neurocomputing, vol. 583, p. 127488, 2024.

[21] P. Zhang, T. Yan, Y. Liu, and H. Lu, "Fantastic Animals and Where to Find Them: Segment Any Marine Animal with Dual SAM," in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2578–2587, 2024.

[22] F. C. Flores, M. F. Carvalho, S. A. Da Silva, L. E. Berton, C. C. Bernardo, A. L. Sevilha, et al., "Segmentation of Hepatocytes Nuclei Using YOLO and Mathematical Morphology," in Proc. 31st Int. Conf. on Systems, Signals and Image Processing (IWSSIP), pp. 1–7, Jul. 2024.

[23] M. Telceken, M. Okuyar, D. Akgun, S. Kacar, and M. S. Vural, "A new data label conversion algorithm for YOLO segmentation of medical images," The European Physical Journal Special Topics, pp. 1–10, 2024.

[24] M. F. Almufareh, M. Imran, A. Khan, M. Humayun, and M. Asim, "Automated brain tumor segmentation and classification in MRI using YOLO-based deep learning," IEEE Access, 2024.

[25] J. Wang, Z. Zhang, B. Dai, K. Zhao, W. Shen, Y. Yin, and Y. Li, "Cow-YOLO: Automatic cow mounting detection based on non-local CSPDarknet53 and multiscale neck," International Journal of Agricultural and Biological Engineering, vol. 17, no. 3, pp. 193–202, 2024.

[26] S. Chanda, Y. N. Kumar, S. Srivastava, R. Rani, M. Shree, and A. K. Mohapatra, "Optimizing facial feature extraction and localization using YOLOv5: An empirical analysis of backbone architectures with data augmentation for precise facial region detection," Multimedia Tools and Applications, pp. 1–22, 2024.