Performance Comparison of YOLOv8, YOLOv5 and YOLOv11 in Nature Image Segmentation

Timothy Harminto*, Dimas F. Najib, Pandu Wicaksono*, and Zahra Nabila Izdihar

Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Email: timothy.harminto@binus.ac.id (T.H.); dimas.najib@binus.ac.id (D.F.N.); pandu.wicaksono005@binus.ac.id (P.W); zahra.izdihar@binus.ac.id (Z.N.I)
*Corresponding author

***Abstract*—Image segmentation is a crucial task in computer vision techniques, serving as a fundamental method for partitioning images into detailed segments that facilitate analysis and retrieval. This paper examines the performance of three YOLO models—particularly YOLOv5, YOLOv8, and YOLOv11—in nature image segmentation, specifically focusing on reptile images. The experiment evaluates accuracy, precision, recall, mean Average Precision (mAP), and computational efficiency using a diverse dataset of reptile images captured under varying environmental conditions.**

In the conducted experiments, we performed comparative tests involving the three models, yielding distinct outputs. Each of these models has its advantages, highlighting the best performance traits of each. YOLOv5 is user-friendly in implementation, YOLOv8 operates effectively without anchors, and YOLOv11 exhibits greater efficiency compared to the other two models. The results indicate that YOLOv11 has made significant advancements in architecture and training methods, establishing it as a versatile option for a range of computer vision tasks.

***Keywords*—computer vision, yolo, image segmentation, nature detection**

I. Introduction

Image segmentation is a computer technique that divides a digital image into pixels by breaking the visual data into specially shaped segments. Image segmentation is a further development of image categorization and object detection. The importance of image segmentation techniques is to speed up and improve image processing. This computer method locates the foreground of the image's main object. Users will only analyze objects in the segmented image. Image segmentation is used to identify the foreground of the main object in the image, making it easier for users to study its location, identify curved lines, and more, and then display it in a single-color image. This ability is crucial in various fields; some examples are in health [1–4], autonomous vehicle [5], and analyzing natural images [6–11]. Analysis of natural images, such as identifying reptiles in their natural habitat, can support important efforts in biodiversity monitoring and conservation studies. However, segmenting animals in the wild is a challenging task due to factors such as natural camouflage, complex backgrounds, varying lighting conditions, and partial occlusion of the subject.

Over the years, image segmentation has been an active area of research, and many different methods have been developed to solve this difficult problem. Every year, researchers come up with new ways to improve accuracy and effectiveness. The You Only Look Once (YOLO) model family [12] has been a powerful solution known for its fast and accurate real-time object recognition. In recent years, this family has changed a lot. YOLOv5 is known for being easy to use and less space-consuming. YOLOv8, the next version of the YOLO series, introduced several improvements, including a detection system that does not require anchors. YOLOv11, the current generation, aims for better efficiency and accuracy by using a better architecture and training method. However, despite the many advances and praises for each model, there is no direct comparison of the performance of these three specific versions (YOLOv5, YOLOv8, and YOLOv11) for reptile segmentation in natural environments in the scientific literature. Although some studies have compared YOLO models, they often focus on different domains or fail to cover all three important iterations. It is unclear what the trade-offs are between accuracy, memory, and computational efficiency for this specific and complex wildlife analysis task. This study aims to conduct a comprehensive performance comparison of YOLOv5, YOLOv8, and YOLOv11 models, with a special emphasis on their application to reptile image segmentation. This analysis is done to address the research gap. This study will compare and evaluate key performance metrics such as accuracy, recall, and mean Average Precision (mAP) by training and testing these algorithms on a diverse reptile dataset in various natural contexts. The goal of this effort is to offer in-depth empirical information on the advantages and disadvantages of each modelling approach. Ultimately, this comparison will serve as a practical reference for researchers and practitioners in selecting the most appropriate model for wildlife image analysis and other comparable computer vision tasks.

Manuscript received March 6, 2025; revised April 30, 2025; accepted June 26, 2025; published September 18, 2025.

doi: 10.18178/joig.13.5.515-520

II. Literature Review

Computer vision has transformed wildlife detection and monitoring by facilitating the automation of animal identification through camera traps and natural imagery [13–15]. Deep learning technology has demonstrated exceptional proficiency in accurately identifying diverse animal species. Automated wildlife monitoring systems employ sophisticated machine learning methodologies to enhance conservation efforts by diminishing manual labor and increasing identification precision.

A. *The Evolution and Comparative Application of YOLO Models*

Several researchers have discussed image segmentation methods, focusing on the YOLOv5, YOLOv8, and YOLOv11 models, each of which offers distinct advantages. YOLOv5, particularly its lightweight version YOLOv5s, is often used as a strong baseline but its performance on small or occluded objects can be limited. Addressing this limitation, a study by Zang *et al.* [16] proposed an enhancement by integrating attention modules, specifically the Efficient Channel Attention (ECA) and Global Attention Mechanism (GAM), into the YOLOv5s architecture. The improved model achieved an accuracy of 71.61%, which is 4.95% higher than the standard YOLOv5s model. Following YOLOv5, the release of YOLOv8 introduced key architectural changes such as an anchor-free design. A study by Reis *et al.* showcased the capabilities of YOLOv8 in real-time flying object detection, confirming its high efficiency and performance. YOLOv8 achieves better mAP50 compared to YOLOv5 across various categories in the RF100 dataset [17]. The evolution continues with YOLOv11, which is designed to further enhance efficiency. Recent research, such as the study by Alif [18], provides a detailed analysis of YOLOv11 for vehicle detection in intelligent transportation systems. This work demonstrates that YOLOv11 introduces architectural improvements such as advanced spatial attention mechanisms and optimized backbone structures that significantly boost detection accuracy and speed, especially for smaller and occluded vehicles, compared to previous YOLO versions [18]. YOLO11 achieved mAP50 of 76.8%, surpassing YOLOv8 (73.9%) and YOLOv10 (74.3%).

B. *Deep Learning for Wildlife Image Segmentation*

Deep learning techniques, especially Convolutional Neural Networks (CNNs), have demonstrated promising outcomes in wildlife detection and species recognition [19, 20]. Beginning with conventional CNN architectures with image processing [21], including AlexNet [22], VGG [23], ResNet [24], and progressing to more sophisticated models such as U-Net, and SegNet [20].

A novel approach involves a two-stage method for wildlife instance segmentation that integrates object detection with contour estimation. This method enhances both species recognition and segmentation accuracy. It employs Few-Shot Object Detection (FSOD) for generating initial bounding boxes and identifying species. Additionally, a deep snake algorithm is utilized for contour estimation, refining the initial bounding box to better conform to the animal's shape. This proposed method exhibits improved performance, particularly when dealing with challenging images [25].

Another technique is to use YOLOv5 with enhancements including the introduction of SPPF, improved feature pyramid structure, GSConv, and VoVGSCSP module, all of which are intended to improve detection speed and accuracy. The proposed enhancements have shown significant improvements in wildlife recognition, accuracy, and speed. The model is designed to be lightweight and can be used on mobile devices, making it practical for real-world applications [26].

C. *Summary*

From the literature, it is evident that while numerous studies have benchmarked YOLO models, most comparisons between YOLOv5, YOLOv8, and YOLOv11 have been conducted in domains such as medical imaging or transportation. A specific, empirical study evaluating their trade-offs for the nuanced task of reptile segmentation in natural environments remains an open area for investigation. This study aims to fill that specific gap.

III. Materials and Methods

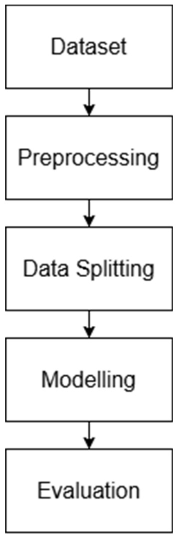

Fig. 1. Diagram of the Experimentation Method

Based on Fig. 1, this research will be conducted in five stages: dataset collection, preprocessing, data splitting, modelling, and evaluation. The evaluation phase will involve assessing the model's performance using various metrics to ensure its accuracy and reliability. This comprehensive approach aims to provide valuable insights and improve overall outcomes in the research process.

A. *Dataset*

This work uses the publicly available reptile dataset from Roboflow Universe, which consists of two classes: "abnormal of reptiles" and "normal of reptiles" [27]. This dataset offers a substantial and varied compilation of annotated images, making it suitable for training and evaluating image segmentation algorithms. Next, we randomly select 200 images from the dataset. This dataset will provide a comprehensive evaluation of the robustness, accuracy, and flexibility of the YOLOv5, YOLOv8, and YOLOv11 algorithms in segmenting reptiles under various backgrounds, lighting conditions, and image qualities, despite the constraints imposed by the data volume. Fig. 2 displays some example images from the dataset. The richness of the dataset ensures that our tests will accurately represent the real-world complexity, thus yielding important insights into the efficacy of these segmentation techniques.

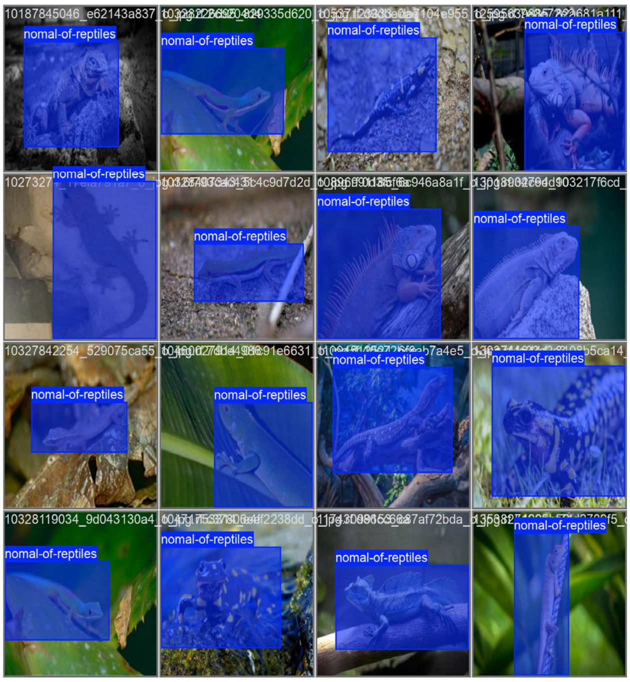

Fig. 2. Example image from dataset.

B. Preprocessing

To make an appropriate comparison of the accuracy of the three algorithms, we will apply the same preprocessing processes to the dataset input for each of the methods. Utilizing the same preprocessing ensures that any increases in accuracy are a result of the algorithms themselves, rather than being the result of variations in the way we prepared the data. The Roboflow picture resizing tool was utilized throughout the training phase to resize each image contained within the dataset to a resolution of 1024 by 1024 pixels.

C. Data Splitting

After the preprocessing step, we separated the dataset into three sets: training, testing, and validation. We assigned 60% of the dataset to training, 20% to testing, and 20% to validation. During the training phase, the testing set will be totally hidden from view, while the training and validation set will be used only for the purpose of training each of the models. With the help of the training and validation set, the models will be able to learn relevant features and patterns, while the testing set will serve as a benchmark to evaluate how well each algorithm performs under standardized conditions. With the help of this division, we can evaluate the models on data that they have never encountered before, which enables us to provide an objective assessment of the generalization performance of each technique.

D. Modelling

All models were trained using their respective Ultralytics frameworks. We utilized a transfer learning methodology by initializing the model with pre-trained weights derived from the COCO dataset. To provide an equitable comparison, identical training settings were uniformly implemented throughout the training of all models (YOLOv5L, YOLOv8L, and YOLOv11L). Each model underwent training for 200 epochs with a batch size of 8. All input images were scaled to a resolution of 1024×1024 pixels. We employed the default training setup of the framework, which featured the AdamW optimizer and an initial learning rate of 0.01. All training procedures were conducted on an NVIDIA Tesla T4 GPU within the Google Colaboratory environment.

E. Model Evaluation

To evaluate the segmentation performance of each algorithm, we will use three key metrics: precision, recall, and mean Average Precision (mAP). These metrics provide a comprehensive view of each algorithm's ability to detect objects accurately and consistently within the images. The terms used are True Positives (TP), False Positives (FP), and False Negatives (FN).

- Precision is calculated as the ratio of TP to the total predicted positives (TP + FP). It quantifies how often the algorithm's detections are correct when it labels an object as positive. This metric is represented by the formula:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

- Recall measures the ratio of TP to actual positives (TP + FN). This metric assesses how well the algorithm captures all relevant instances of the objects in the dataset. The formula for recall is:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

- Mean Average Precision (mAP) is a metric used to measure the performance of a model whose task is to detect an object and retrieve information from images. Mean Average Precision (mAP) itself is a performance metric that is often used in evaluating a machine learning model. Mean Average Precision (mAP) combines precision and recall across various threshold values to summarize the algorithm's overall segmentation accuracy:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad (3)$$

i = class of object being detected
AP_i = Average Precision of class i
n = the number of classes.

IV. Result and Discussion

Different learning characteristics for each model are shown in Table I. As epochs progress, YOLOv5 exhibits a steady learning trajectory, with improvements in both precision and mAP metrics. There appears to be trouble maintaining performance balance, as evidenced by the recall metric's instability, which dropped from 0.775 at epoch 100 to 0.625 at epoch 180 before slightly recovering. On the other hand, YOLOv8 exhibits quick convergence, reaching a high precision performance of 0.748 by the 100th epoch. After this point, performance tends to plateau and, in some metrics, declines slightly, indicating that further training provides little benefit and that the model has reached its peak performance rather early. On the other hand, up to epoch 200, YOLOv11 shows the best training profile, with steady and consistent improvements in all metrics.

TABLE I. Performance of YOLOv5, YOLOv8 and YOLOv11 during Training

Model	Metrics				
	Epoch	Precision	Recall	mAP50	mAP95
YOLOv5	100	0.34301	0.775	0.4514	0.1643
	180	0.54135	0.625	0.57418	0.20731
	200	0.57232	0.675	0.63311	0.23331
YOLOv8	100	0.74838	0.7	0.66832	0.20179
	180	0.72704	0.75	0.6327	0.21612
	200	0.73458	0.725	0.60152	0.2058
YOLOv11	100	0.35694	0.6	0.47232	0.15011
	180	0.62134	0.65648	0.58909	0.20183
	200	0.68307	0.725	0.67871	0.25188

More complex results are obtained by comparing peak performance at the end of training (epoch 200). With the highest mAP50 of 0.67871 and the highest mAP95 of 0.25188, YOLOv11 showed exceptional performance. Interestingly, at epoch 200, YOLOv8 had the lowest mAP50 score (0.60152) out of the three models, despite having the highest precision (0.73458). The results show a clear trade-off: YOLOv8 has a low false positive rate and high detection accuracy, but it is not as good as YOLOv11 at identifying all relevant objects. While YOLOv11 requires more epochs to reach its superior peak potential, YOLOv8's rapid convergence suggests that this model can achieve competitive performance more quickly.

TABLE II. Performance of YOLOv5, YOLOv8 and YOLOv11 during Testing

Model	Metrics				
	Inference Time (ms)	Precision	Recall	mAP50	mAP95
YOLOv5	85.0	0.549	0.7	0.543	0.22
YOLOv8	119.1	0.629	0.7	0.617	0.194
YOLOv11	97.4	0.559	0.725	0.706	0.196

Table II presents the outcomes of the testing conducted with the test data. YOLOv11 exhibits the highest mAP50 accuracy, recorded at 0.706. Nonetheless, considerations of computational efficiency mitigate this elevated performance. YOLOv5 is the fastest model, exhibiting an inference time of 85.0 ms, though it has the lowest mAP50 accuracy at 0.543. In contrast, the most accurate model, YOLOv11 (97.4 ms), exhibits slower performance than YOLOv5, yet is considerably faster than YOLOv8 (119.1 ms), the slowest model in this evaluation.

A trade-off exists between precision and recall. YOLOv8 demonstrates the highest precision at 0.629, indicating a low incidence of false detections. YOLOv11 demonstrates the highest recall at 0.725, indicating its superior capability in detecting all objects. In the mAP95 metric, YOLOv5 achieved the highest score of 0.22, indicating that when this model successfully detects an object, its prediction box localization is frequently accurate and closely aligned with the original object.

This finding underscores that the selection of the optimal model is contingent upon the prioritization criteria specific to the application, including factors such as the importance of overall accuracy versus inference speed, or the need to minimize false positives versus achieving precise image localization.

To further contextualise these findings, we compared them with results from a broader benchmark study. Jegham *et al.* (2024) [28] conducted a full benchmark test of several versions of YOLO, from YOLOv3 to YOLOv12, on various object recognition tasks, including tasks with objects of various sizes and aspect ratios. YOLOv11 was one of the most consistent models in this study in terms of balancing accuracy (mAP50 of 0.893 and mAP50-95 of 0.795) and computational efficiency (inference time as low as 2.4 ms for model "m"). This conclusion is consistent with the results in Table II, which show that YOLOv11 has the largest mAP50, although its inference time is slightly longer than YOLOv5.

A limitation of this study is that the model was only evaluated based on a single training run, consequently although the results show clear performance trends, we cannot formally claim statistical significance for the observed differences. Furthermore, the reliance on a relatively small dataset and focus on a single object class (reptiles), which may limit the ability to generalize the findings to a more diverse domain. Additionally, we did not perform hyperparameter optimization, which may affect the peak performance of each model due to the limited computing power available. Nevertheless, this study provides an initial basis for further research in this area.

V. Conclusion

This study has conducted a thorough performance evaluation of YOLOv5, YOLOv8, and YOLOv11 models in the context of reptile image segmentation inside a realistic setting. The primary findings indicate that no singular model outperforms in all criteria. YOLOv11 demonstrated the highest overall accuracy (mAP50), although YOLOv8 excelled in precision, and YOLOv5 provided the fastest inference speed. This study's primary contribution is the validation of a distinct trade-off among accuracy, precision, and computing efficiency, providing a realistic framework for researchers to select the design that optimally aligns with their application requirements.

This investigation has many limitations that should be acknowledged. The primary restriction is that the assessment relies on a solitary training iteration, therefore precluding formal assertions regarding the statistical significance of the performance disparities. The utilization of a relatively limited dataset and concentration on a singular object class constrains the generalizability of these findings to a broader context.

Subsequent study may investigate the efficacy of YOLOv11, YOLOv8, and YOLOv5 by evaluating their effectiveness across various factors, such as reduced epochs, differing learning rates, and larger, more intricate datasets. Such research can yield profound insights into the trade-offs among model efficiency, accuracy, and training duration. A possible avenue for research is to assess these models on datasets containing numerous objects for segmentation, which would evaluate their capacity to manage overlapping instances, intricate object interactions, and varied background conditions. Furthermore, evaluating these models on specialized datasets tailored for industries, such as medical imaging, autonomous driving, or agricultural analysis, helps elucidate their strengths and drawbacks in specific applications. Integrating criteria such as precision, recall, mean Average Precision (mAP), and computing efficiency might enhance the assessment of their effectiveness in practical applications.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Dimas F. Najib wrote the abstract, literature review and experimentation method. Timothy Harminto performed all experiments, provided result analysis and wrote the conclusion for this paper. Pandu Wicaksono provided guidance and editorial services for this paper. Zahra N. Izdihar proofread the paper and provided editorial services. All authors had approved the final version.

Acknowledgements

In the preparation of this manuscript, the authors utilized a generative artificial intelligence tool (Quillbot) as a language editing assistant. The role of the AI was strictly limited to improving grammar, refining sentence structure, and enhancing the overall clarity and academic tone of the text.

References

[1] I. B. Ulitin, M. Barulina, and M. Velikanova, "Using the STEGO neural network for scintigraphic image analysis," *INTELS'22*, 2023. https://www.mdpi.com/2673-4591/33/1/5

[2] A. M. Prakoso, A. W. Khair, and B. Pumama, "YOLOv9: Advancing automated brain tumor detection in MRI scans," in *Proc. 2024 12th International Conference on Information and Communication Technology (ICoICT)*, 2024, pp. 572–577. https://ieeexplore.ieee.org/abstract/document/10698488/

[3] A. Awad and S. A. Aly, "Early diagnosis of acute lymphoblastic leukemia using YOLOv8 and YOLOv11 deep learning models," arXiv preprint, arXiv:2410.10701, 2024. https://arxiv.org/abs/2410.10701

[4] C.-T. Chien, R. Ju, K.-Y. Chou, and J.-S. Chiang, "YOLOv9 for fracture detection in pediatric wrist trauma X-ray images," arXiv Print, arXiv:2403.11249, 2024. doi: 10.48550/arXiv.2403.11249

[5] D. Bogdoll, N. Ollick, T. Joseph, and J. M. Zöllner, "UMAD: Unsupervised mask-level anomaly detection for autonomous driving," arXiv Print, arXiv:2406.06370, 2024.

[6] M. Zhang, F. Gao, W. Yang, and H. Zhang, "Wildlife object detection method applying segmentation gradient flow and feature dimensionality reduction," *Electronics (Basel)*, vol. 12, no. 2, 2023. doi: 10.3390/electronics12020377

[7] M. Zeppelzauer, "Automated detection of elephants in wildlife video," *EURASIP J. Image Video Process*, vol. 2013, no. 1, p. 46, 2013. doi: 10.1186/1687-5281-2013-46

[8] C. Zhang and J. Zhang, "DJAN: Deep joint adaptation network for wildlife image recognition," *Animals*, vol. 13, no. 21, 2023. doi: 10.3390/ani13213333

[9] D. Lu and Y. Wang, "MAR-YOLOv9: A multi-dataset object detection method for agricultural fields based on YOLOv9," *PLoS One*, vol. 19, 2024. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0307643

[10] N. Nikbakhsh, Y. Baleghi, and H. Agahi, "A novel approach for unsupervised image segmentation fusion of plant leaves based on G-mutual information," *Mach. Vis. Appl.*, vol. 32, 2020. https://link.springer.com/article/10.1007/s00138-020-01130-0

[11] H. Zang *et al.*, "Detection method of wheat spike improved YOLOv5s based on the attention mechanism," *Front Plant Sci.*, vol. 13, 2022. https://www.frontiersin.org/articles/10.3389/fpls.2022.993244/full

[12] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2015. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[13] S. Mustafic, D. Dachs, R. Prüller, F. Schöggl, and R. Perko, "Improving wildlife management with AI: Species detection and classification from camera trap data," in *CamTraps*, 2024. https://ceur-ws.org/Vol-3786/paper2.pdf

[14] M. Mulero-Pázmány *et al.*, "Addressing significant challenges for animal detection in camera trap images: A novel deep learning-based approach," *Sci. Rep.*, vol. 15, no. 1, 16191, 2025. doi: 10.1038/s41598-025-90249-z

[15] M. Tan *et al.*, "Animal detection and classification from camera trap images using different mainstream object detection architectures," *Animas (Basel)*, vol. 12, 2022. https://www.mdpi.com/2076-2615/12/15/1976

[16] H. Zang *et al.*, "Detection method of wheat spike improved YOLOv5s based on the attention mechanism," *Front Plant Sci.*, vol. 13, 2022. doi: 10.3389/fpls.2022.993244

[17] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," arXiv Print, arXiv:2305.09972, 2023. https://arxiv.org/abs/2305.09972

[18] M. A. R. Alif, "YOLOv11 for vehicle detection: Advancements, performance, and applications in intelligent transportation systems," arXiv Print, arXiv:2410.22898, 2024. https://arxiv.org/abs/2410.22898

[19] Z. Xu, T. Wang, A. K. Skidmore, and R. Lamprey, "A review of deep learning techniques for detecting animals in aerial and satellite images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 128, 103732, 2024. doi: 10.1016/j.jag.2024.103732

[20] A. Lumini, G. B. Rozendo, M. Dadi, and A. Franco, "Comparison of CNN and transformer architectures for robust cattle segmentation in complex farm environments," in *Proc. International Conference on Pattern Recognition Applications and Methods*, 2025. https://www.scitepress.org/Papers/2025/131764/131764.pdf

[21] S. Faizal and S. Sundaresan, "Wild animal classifier using CNN," arXiv Print, arXiv:2210.07973, 2022. https://arxiv.org/abs/2210.07973

[22] R. Adhvaryu, Komil. B. Vora, and D. G. Kamdar, "Summarizing animal video using pre-trained Alexnet and deep learning," *SSRN Electronic Journal*, 2019. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3462619

[23] H. Nguyen *et al.*, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring," in *Proc. 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2017, pp. 40–49. https://ieeexplore.ieee.org/abstract/document/8259762/

[24] S. Sharma, S. Dhakal, and M. Bhavsar, "Transfer learning for wildlife classification: Evaluating YOLOv8 against DenseNet, ResNet, and VGGNet on a Custom dataset," *Journal of Artificial Intelligence and Capsule Networks*, 2024. https://arxiv.org/abs/2408.00002

[25] 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, 2022. doi: 10.3390/ani12151980

[26] M. Zhang, F. Gao, W. Yang, and H. Zhang, "Wildlife object detection method applying segmentation gradient flow and feature dimensionality reduction," *Electronics (Basel)*, vol. 12, no. 2, 2023. doi: 10.3390/electronics12020377

[27] CapStone. (Nov. 2023). reptile2 dataset. *Roboflow* [Online]. Available: https://universe.roboflow.com/capstone-habql/reptile2

[28] N. Jegham, C. Y. Koh, M. F. Abdelatti, and A. M. Hendawi, "YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions," *Yolo11, and Their Previous Versions,* 2024. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5175639