

Animal Species Classification Using Deep Learning

Rizqi Andhika Pratama

Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, Indonesia

18222118@std.stei.itb.ac.id

Khayla Belva Annandira

Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, Indonesia

18222138@std.stei.itb.ac.id

Mohamad Maulana Firdaus R.

Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

Bandung, Indonesia

18222140@std.stei.itb.ac.id

Abstract—Automatic classification of animal species from images is an important task in biodiversity research, ecological monitoring, and educational tools. This study evaluates the performance of five state-of-the-art deep learning models—VGG16, ResNet50, EfficientNet-B3, ViT-B16, and YOLOv11-cls—on the Animals10 dataset. EfficientNet-B3 achieved the highest classification accuracy (98.38%) and outperformed other models across most evaluation metrics. In addition, YOLOv11-cls demonstrated competitive performance with the fastest inference time, making it suitable for real-time applications. The best model was deployed in a web-based interface using FastAPI and Next.js to allow users to interactively classify animal species. This research provides a benchmark for modern animal classification systems and demonstrates the potential for integrating deep learning into accessible applications for education and conservation.

Index Terms—Animal Classification, Deep Learning, EfficientNet, YOLOv11, Vision Transformer, Convolutional Neural Network, Computer Vision, Web Deployment, FastAPI, Next.js

I. INTRODUCTION

The classification of animal species is a fundamental task in biology, supporting research in biodiversity, conservation, and ecological monitoring. Traditionally, species identification has relied on manual observation and expert knowledge, which can be time-consuming and prone to human error, especially when dealing with large datasets or subtle morphological differences among species.

With the advancement of computing technology, automated image-based classification methods have become increasingly popular. Deep learning, particularly Convolutional Neural Networks (CNNs), has demonstrated remarkable performance in various computer vision tasks, including object detection and image classification. These methods enable the extraction of complex features from images, allowing for more accurate and scalable animal classification compared to traditional approaches.

Recent studies have explored the application of various deep learning architectures—such as VGG, ResNet, EfficientNet, Vision Transformers (ViT), and YOLO—across multiple wildlife and biodiversity datasets, yielding promising results. However, challenges remain in achieving consistently high accuracy across diverse animal classes, particularly when datasets are imbalanced or contain species with visually similar features. Moreover, comparative evaluations on standardized datasets, such as Animals10, remain limited.

In addition to model performance, the practical usability of deep learning models also depends on how seamlessly they can be integrated into real-time systems. Therefore, this study also explores the development of a full-stack application using Next.js and FastAPI to facilitate real-time animal classification through a web interface.

Research Question: How do different state-of-the-art deep learning architectures, including CNN-based models and YOLO-based classifiers, perform in classifying animal species from the Animals10 dataset?

This study also includes a prototype interface built using Next.js (frontend) and FastAPI (backend) to demonstrate the deployment of trained models in a real-time classification setting.

Purpose and Contribution: This study aims to systematically evaluate and compare the performance of several state-of-the-art deep learning models—including VGG16, ResNet50, EfficientNet-B3, Vision Transformer (ViT), and YOLOv11-cls—on the Animals10 image dataset. The main contributions of this research are:

- Providing a comprehensive benchmark of modern deep learning models for animal image classification.
- Analyzing the strengths and limitations of each model in the context of biological image data.
- Offering insights and recommendations for future research and practical applications in automated animal classification.
- Prototyping a practical implementation using a modern full-stack interface: a web-based frontend using Next.js and a FastAPI backend for model deployment and prediction, enabling real-time species classification.

II. RECENT STATE-OF-THE-ART IN ANIMAL IMAGE CLASSIFICATION

In recent years, a wide range of deep learning models have been proposed to address the challenge of animal species classification from images, especially in ecological and wildlife monitoring contexts. This section highlights key developments involving CNN-based architectures, Vision Transformers (ViT), and object detection models such as YOLO.

CNN-Based Architectures. Convolutional Neural Networks (CNNs) remain widely used for image classification due to their ability to capture hierarchical spatial features.

In [1], Edmonds et al. evaluated multiple CNNs, including EfficientNetB7, DenseNet, and ResNet, for classifying ten animal species, finding that EfficientNetB7 provided the best trade-off between accuracy and computational cost. Similarly, Islam et al. [2] compared VGG16, ResNet50, and a custom CNN, showing that pretrained models via transfer learning significantly outperform self-trained ones on camera trap images, achieving up to 87% accuracy.

Transformer-Based Architectures. Transformer-based models such as the Vision Transformer (ViT) have demonstrated strong performance in image classification by leveraging self-attention mechanisms. Rezapour [3] trained a ViT on animal image datasets and achieved 89% classification accuracy. The study underlined ViT’s ability to capture global image context, making it highly effective for distinguishing animal classes with subtle visual differences.

YOLO-Based Real-Time Detection. The YOLO family of models has gained popularity in wildlife applications due to its real-time detection capability. Thangaraj et al. [4] compared YOLOv3 and YOLOv5 for animal tracking in camera trap datasets, showing that YOLOv5-m reached 81% detection accuracy while maintaining fast inference. Sharma et al. [5] extended this with YOLOv11, which achieved a validation F1-score of 96.5% when classifying endangered species, outperforming other CNN-based models.

Transfer Learning and Comparative Performance. Transfer learning has become an essential tool to improve model performance on small ecological datasets. Boukar et al. [6] evaluated VGG19, GoogLeNet, ResNet50, and DenseNet121, showing GoogLeNet yielded the highest classification accuracy at 87%. These findings confirm the effectiveness of using pretrained weights and model fine-tuning for challenging wildlife datasets.

Interpretable Deep Learning in Ecology. In addition to performance, interpretability is crucial in ecological monitoring. Miao et al. [7] explored visual feature attribution techniques such as Grad-CAM and Guided Backpropagation to interpret CNN decisions in wildlife classification, achieving 87.5% accuracy on 20 African species. This helps validate model predictions and supports trust in AI-assisted biodiversity research.

These works collectively demonstrate the significant progress in animal classification tasks through a variety of modern deep learning approaches, each suited to different contexts—from static image classification to real-time multi-object detection in the wild.

III. METHOD

A. Dataset Description

The Animals10 dataset [8] consists of approximately 28,000 medium-quality images of animals, divided into 10 categories: dog, cat, horse, spider, butterfly, chicken, sheep, cow, squirrel, and elephant. The images were collected from Google Images and manually verified, with a small portion of noisy data intentionally included to simulate real-world conditions. The

dataset is organized into separate folders for each class, with the number of images per class ranging from around 1,400 to nearly 5,000.

For this study, the original Italian class names were mapped to Indonesian for consistency. Table I shows the number of images per class and their distribution after an 80:20 train-test split.

TABLE I
NUMBER OF IMAGES PER CLASS IN THE ANIMALS10 DATASET AND THEIR TRAIN-TEST SPLIT.

Class	Total	Train (80%)	Test (20%)
Anjing	4863	3890	973
Laba-laba	4821	3856	965
Ayam	3098	2478	620
Kuda	2623	2098	525
Kupu-kupu	2112	1689	423
Sapi	1866	1492	374
Tupai	1862	1489	373
Domba	1820	1456	364
Kucing	1668	1334	334
Gajah	1446	1156	290
Total	26179	20938	5241

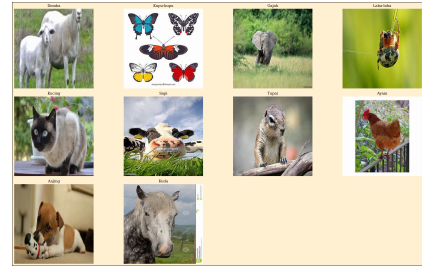


Fig. 1. Example images from the Animals10 dataset.

B. Model Architectures

This study evaluates several state-of-the-art deep learning models for image classification:

1) **VGG16:** VGG16 is a deep convolutional neural network architecture introduced by Simonyan and Zisserman [9]. It consists of 16 layers with small 3×3 convolution filters and is known for its simplicity and effectiveness in image classification tasks.

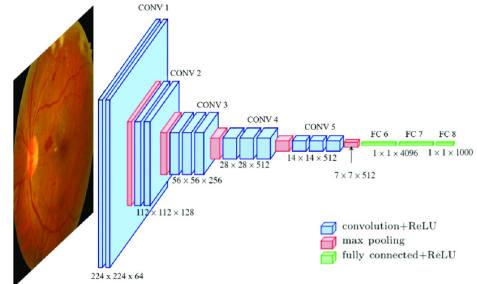


Fig. 2. VGG16 architecture overview [9].

2) *ResNet50*: ResNet50, introduced by He et al. [10], is a 50-layer deep convolutional neural network that utilizes residual connections to enable the training of very deep networks, effectively mitigating the vanishing gradient problem.

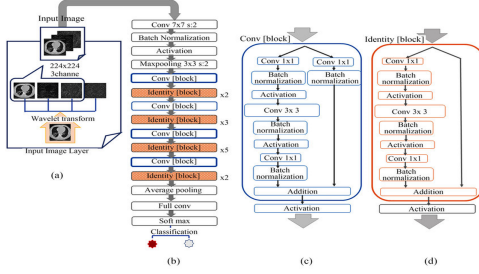


Fig. 3. ResNet50 architecture overview [10].

3) *EfficientNet-B3*: EfficientNet-B3 is part of the EfficientNet family [11], which scales network width, depth, and resolution using a compound coefficient. EfficientNet-B3 achieves high accuracy with fewer parameters compared to traditional CNNs.

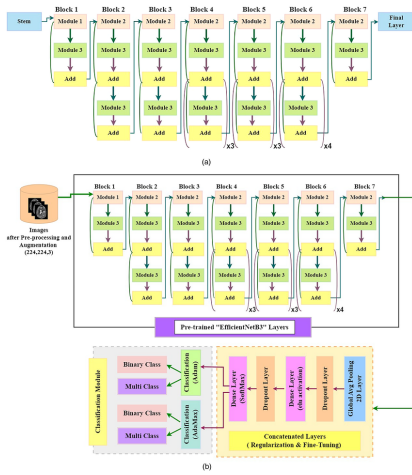


Fig. 4. EfficientNet-B3 architecture overview [11].

4) *Vision Transformer (ViT)*: The Vision Transformer (ViT) [12] applies the transformer architecture, originally designed for natural language processing, to image classification by splitting images into patches and processing them as a sequence.

5) *YOLOv11*: YOLOv11 [13] is a real-time object detection and classification model. In this study, the YOLOv11-cls variant is used for image-level classification, leveraging its speed and efficiency.

C. System Workflow

Figure 7 illustrates the end-to-end workflow of the system. The process begins with dataset preparation, followed by preprocessing and model training. After evaluation, the best-performing model is deployed through a FastAPI backend and integrated into a Next.js web interface for real-time prediction.

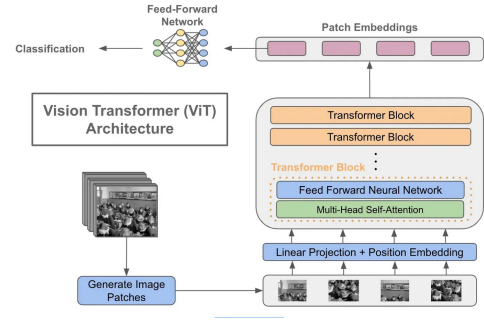


Fig. 5. Vision Transformer (ViT) architecture overview [12].

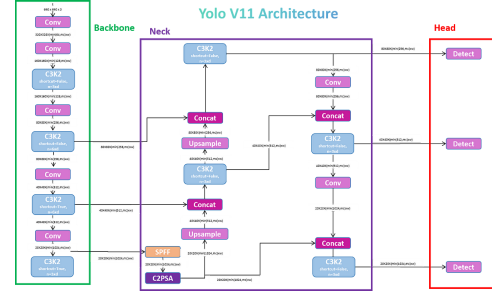


Fig. 6. YOLOv11 architecture overview [13].

D. Computational Environment

All experiments were conducted using Kaggle Notebooks (Kaggle Kernels) with the following specifications:

- **CPU**: Intel(R) Xeon(R) CPU @ 2.20GHz, 2 cores, 4 threads
- **RAM**: 32 GB
- **GPU**: NVIDIA Tesla P100, 16 GB
- **Platform**: Kaggle Notebook (Ubuntu 20.04-based kernel)
- **Libraries**: Python 3.10, PyTorch 2.0, torchvision, timm, ultralytics

E. Training Configuration

Each model was trained using the following configuration:

- **Optimizer**: Adam
- **Learning Rate**: 0.0001
- **Batch Size**: 32
- **Epochs**: 20 (early stopping counts 3 applied)
- **Loss Function**: Cross Entropy Loss

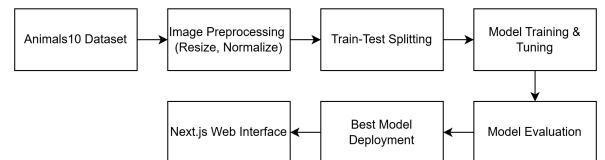


Fig. 7. Overall system workflow from dataset preparation to interface deployment.

F. Evaluation Metrics

To assess the model’s classification performance, we used the following evaluation metrics:

- **Accuracy:** The ratio of correctly predicted samples over all predictions.
- **Precision:** The proportion of true positives among all positive predictions.
- **Recall:** The proportion of true positives among all actual positives.
- **F1-Score:** The harmonic mean of precision and recall, useful for imbalanced classes.
- **Confusion Matrix:** To visualize per-class performance and detect misclassifications.

G. Model Deployment

After training, the best-performing model was exported as a PyTorch ‘.pt’ file and served through a FastAPI backend. The API exposes an endpoint ‘/predict’, which receives an image file and returns a predicted class and confidence score. The backend supports GPU acceleration via CUDA when available.

H. Interface Design

The EyeNimal application provides a user-friendly interface for animal image classification. Users can upload or capture an image of an animal, which is then processed by the trained deep learning model. The interface displays the predicted species, scientific name, and additional information about the animal.

The application is built as a full-stack system. The frontend is implemented using **Next.js**, a React-based framework, while the backend is powered by **FastAPI**, a modern Python web framework for serving the trained model via REST API. Image data is sent from the frontend to the backend in real-time, and the prediction response is returned and displayed instantly on the UI.

Figures 8 show several key screens of the application, including the home page, image upload, prediction result, and animal detail page.

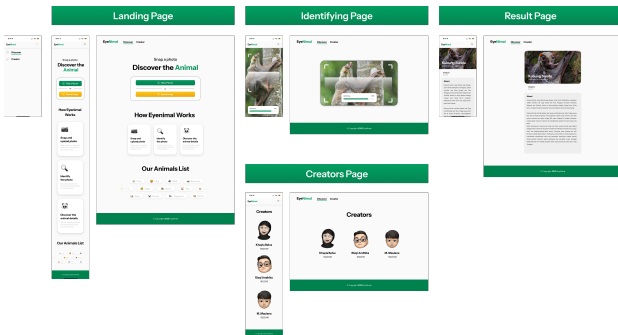


Fig. 8. Key interface screens of the EyeNimal application.

I. Code Availability

The full source code of this project—including model training scripts, evaluation pipelines, frontend and backend implementation—is available on GitHub:

<https://github.com/qiewi/eye-nimal>

The repository is organized into separate folders for models/, training/, frontend/ and backend/.

IV. RESULTS AND DISCUSSION

A. Quantitative Results

The evaluation results show that EfficientNet-B3 outperforms all other models with the highest validation accuracy (98.38%) and the best precision, recall, and F1-score. It also achieves the lowest validation loss, indicating excellent generalization. While YOLOv8-clas does not report full metrics, it offers competitive accuracy and the fastest inference time (1.20 ms), making it ideal for real-time use. ResNet50 and ViT-B16 perform well but with slightly higher loss, and VGG16 shows signs of overfitting.

B. Training Behavior and Convergence

Figure 9 shows the training and validation curves for all models. EfficientNet-B3 converges smoothly with minimal overfitting, while others like VGG16 and ViT show more fluctuation in validation loss.

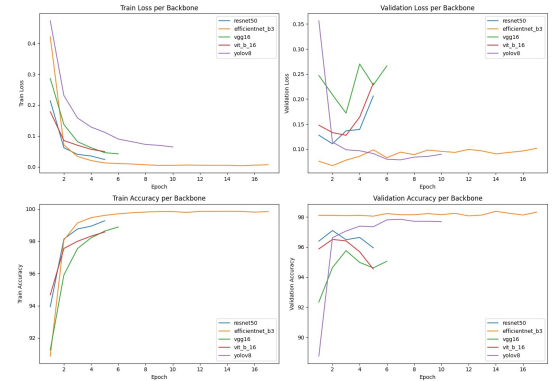


Fig. 9. Training and validation loss and accuracy curves for all models.

YOLOv11 shows relatively high training loss, but its validation accuracy improves rapidly and remains competitive. This highlights its efficient architecture, despite being primarily designed for object detection.

C. Qualitative Results

Figure 10 displays sample predictions from EfficientNet-B3. Correct classifications are consistent across diverse species. Misclassifications occur mostly between animals with similar shapes or poses, such as:

- *Domba* misclassified as *Sapi*
- *Tupai* misclassified as *Domba*
- *Kucing* misclassified as *Anjing*

To further investigate how models “see,” Grad-CAM visualizations in Figure 11 show the activation regions during

TABLE II
VALIDATION PERFORMANCE AND LOSS METRICS OF ALL MODELS ON THE ANIMALS10 DATASET.

Model	Train Loss	Val Loss	Val Acc	Val Prec	Val Recall	Val F1	Train Time	Infer Time (ms)
EfficientNet-B3	0.0053	0.0905	98.38%	98.30%	98.25%	98.27%	61m 46s	12.93
YOLOv11-cls	0.0815	0.0785	97.84%	—	—	—	17m 49s	1.20
ResNet50	0.0620	0.1107	97.10%	96.89%	96.86%	96.84%	17m 1s	7.21
ViT-B16	0.0853	0.1330	96.50%	95.87%	96.22%	95.99%	39m 25s	10.90
VGG16	0.0816	0.1719	95.76%	95.63%	95.15%	95.33%	24m 18s	5.14

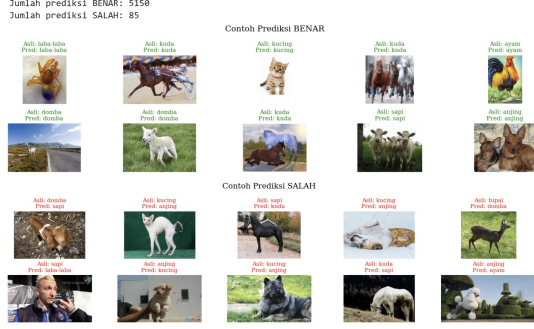


Fig. 10. Examples of correct and incorrect predictions by EfficientNet-B3.

prediction. Correct predictions typically focus on semantically relevant areas (e.g., animal face, body shape), while incorrect predictions highlight non-discriminative regions (e.g., background or partial limbs). This validates the hypothesis that spatial attention plays a critical role in robust classification and suggests potential for integrating attention-guided training in future work.

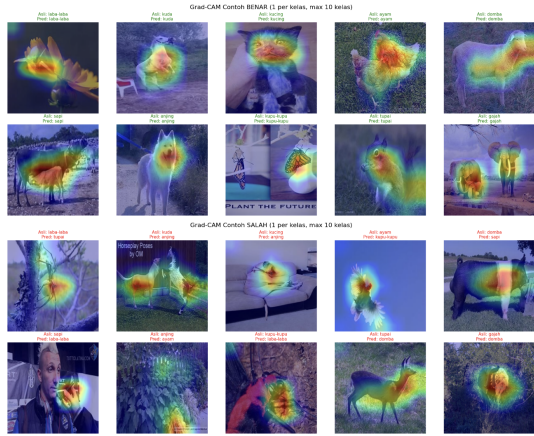


Fig. 11. Grad-CAM visualization of correctly (top) and incorrectly (bottom) classified images using EfficientNet-B3. The highlighted regions indicate the focus of the model during classification.

D. Visual Similarity Analysis

A dendrogram based on EfficientNet-B3's feature space (centroids of class embeddings) is shown in Figure 12. The hierarchical clustering aligns well with biological categories:

- **Kuda, Domba, Sapi** clustered as large mammals
- **Kupu-kupu** and **Ayam** grouped due to winged features

- **Kucing, Anjing, and Tupai** formed a small mammal cluster

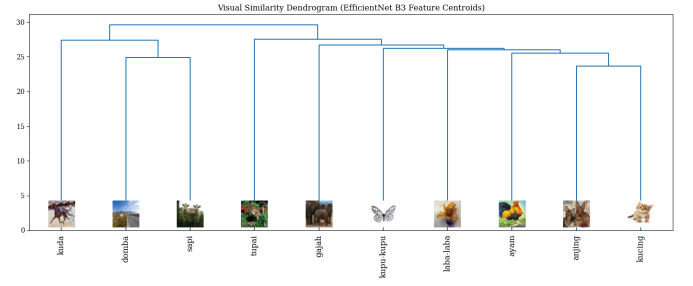


Fig. 12. Visual similarity dendrogram using EfficientNet-B3 feature embeddings.

E. Biological Discussion

The results reflect the importance of visual morphology in species classification:

- *Laba-laba* and *Kupu-kupu* are consistently well-classified due to distinctive anatomical structures (e.g., legs, wings).
- Errors often occur in classes with overlapping characteristics, e.g., quadrupeds with similar fur or size.
- Despite no biological knowledge given to the model, embeddings demonstrate taxonomic-like groupings.

This demonstrates that convolutional and transformer-based models can implicitly learn biologically relevant distinctions.

F. Confusion Matrix Analysis

Figure 13 The confusion matrix reveals that most misclassifications occur between visually similar mammals, such as *sheep-cow* or *squirrel-dog*, while classes with distinct appearances like *spider*, *butterfly*, and *chicken* are rarely confused. This underscores the importance of morphological distinctiveness in visual classification.

G. System Deployment Perspective

The fastest model, yolov11-cls, was deployed via FastAPI and accessed through a Next.js frontend in the EyeNimal interface. This enables real-time prediction with uploaded animal images. An example interface view is shown in Figure 14.

CONCLUSION

This study evaluated and compared the performance of five state-of-the-art deep learning models—VGG16, ResNet50, ViT-B16, EfficientNet-B3, and YOLOv11-cls—on the Animals10 dataset for multi-class animal image classification.

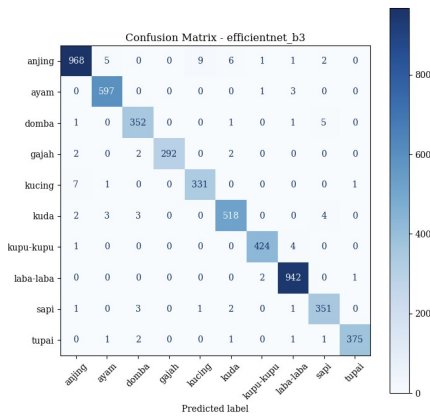


Fig. 13. Confusion matrix for EfficientNet-B3 predictions.

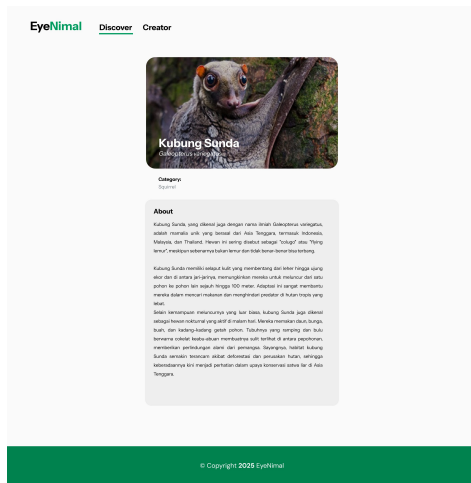


Fig. 14. Real-time prediction result in the EyeNimal web interface.

Based on the experimental results, the following key conclusions can be drawn:

- **EfficientNet-B3** consistently achieved the highest accuracy, precision, recall, and F1-score, while also demonstrating smooth convergence and low validation loss, making it the most reliable model overall for animal classification tasks.
- **YOLOv11-cls**, although initially designed for object detection, achieved competitive classification accuracy with the fastest inference time (1.20 ms/image), indicating its suitability for real-time or edge deployment scenarios.
- Deep learning models were able to implicitly learn biologically relevant visual features (e.g., shape, limb count, texture), as shown by class clustering and confusion matrix patterns that align with taxonomic groupings.
- Through Grad-CAM analysis and visual clustering via dendrograms, the models were shown to implicitly learn biologically meaningful features, such as morphology and structure, without explicit supervision
- The deployment of yolov11-cls in a full-stack application using FastAPI and Next.js demonstrates the feasibility of

integrating accurate animal classification into interactive web interfaces, potentially benefiting education, conservation, and citizen science platforms.

FUTURE WORK

While the models performed well, several areas remain for future improvement:

- **Dataset enrichment:** The Animals10 dataset has class imbalance and limited intra-class variation. Expanding the dataset with more diverse and high-resolution samples can improve generalization.
- **Domain-specific adaptation:** Fine-tuning models for wildlife imagery or real camera-trap data can improve ecological monitoring applications.
- **Mobile deployment:** Compressing or quantizing the model for on-device inference (e.g., TensorFlow Lite, ONNX) could support offline use in field settings with limited connectivity.
- **Multimodal integration:** Combining image features with metadata (e.g., location, time, sound) could enhance prediction accuracy and context-aware classification.

Overall, this research provides a benchmark and practical foundation for automated animal classification systems using modern deep learning techniques, while opening avenues for future enhancements in both accuracy and usability.

REFERENCES

- [1] A. Edmonds, C. A. Sadeghi, and V. Brahmatewari, "Counting Convolutional Neural Networks for Classification for Wildlife Conservation," Stanford University, 2024.
- [2] S. B. Islam et al., "Animal species recognition with deep convolutional neural networks from ecological camera trap images," *Animals*, vol. 13, no. 1526, 2023.
- [3] M. Rezapour, "Animal Image Classification Using the Vision Transformer Model," Preprint, 2024.
- [4] R. Thangaraj et al., "Automatic Detection and Classification of Wild Animal Species Using YOLO Models," in *Proc. 3rd Int. Conf. on Pervasive Computing and Social Networking (ICPCSN)*, 2023, pp. 315–320.
- [5] S. Sharma, S. Dhakal, and M. Bhavsar, "Transfer Learning for Wildlife Classification: Evaluating YOLOv8 against DenseNet, ResNet, and VGGNet," *J. Artif. Intell. Capsule Netw.*, vol. 6, no. 4, pp. 415–435, 2024.
- [6] M. M. Boukar et al., "Improving the Accuracy of Animal Species Classification in Camera Trap Images Using Transfer Learning," in *Proc. Int. Conf. on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, 2024.
- [7] Z. Miao et al., "Insights and approaches using deep learning to classify wildlife," *Sci. Rep.*, vol. 9, no. 8137, 2019.
- [8] A. Corrado, "Animals-10: Image dataset of 10 animal classes," Kaggle, 2019. [Online]. Available: <https://www.kaggle.com/datasets/alessiocorrado99/animals10/>
- [9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in **Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)**, 2016, pp. 770–778.
- [11] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in **Proc. Int. Conf. on Machine Learning (ICML)**, 2019, pp. 6105–6114.
- [12] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in **Proc. Int. Conf. on Learning Representations (ICLR)**, 2021.
- [13] G. Jocher et al., "YOLO by Ultralytics," [Online]. Available: <https://github.com/ultralytics/ultralytics>, Accessed: May 28, 2025.