Received: August 7, 2024. Revised: September 18, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Wildlife Species Classification from Camera Trap Images Using Fine-tuning EfficientNetV2

Thanh-Nghi Doan[1,2]* **Duc-Ngoc Le-Thi[1,2]**

[1]*Faculty of Information Technology, An Giang University, An Giang, Vietnam*
[2]*Vietnam National University, Ho Chi Minh City, Vietnam*
* Corresponding author's Email: dtnghi@agu.edu.vn

Abstract: Camera traps are a valuable tool for wildlife research and conservation, but wildlife species classification in camera trap imagery is challenging due to the variation in species appearance, pose, and lighting conditions. This study explores the use of transfer learning and fine-tuning to develop a robust deep convolutional neural network model for wildlife species classification from camera trap images. To prevent overfitting, data augmentation techniques were applied during the image pre-processing stage. ResNet-50 and various EfficientNetV2 variants have been evaluated, and the EfficientNetV2-L model emerged as the top performer. Fine-tuning methods were then applied to the EfficientNetV2-L model to further improve its performance. Experimental results show that the fine-tuned EfficientNetV2-L model outperformed other methods with an accuracy of 88.822%, a precision of 86.941%, a recall of 87.638%, and an F1-score of 87.193% on a held-out test set, demonstrating its effectiveness for wildlife species classification from camera trap images.

Keywords: Wildlife species classification, Camera trap imagery, Deep convolutional neural network, Transfer learning, Fine-tuning.

1. Introduction

Camera traps revolutionize wildlife research and conservation, enabling non-invasive monitoring of wildlife species in their natural habitats [1, 2]. Capturing an extensive array of images, this potent tool has the potential to yield a wealth of information about the presence of an animal in a carefully chosen study area [3], its population size, and interactions within the community [4]. Researchers can remotely amass biographical and crucial evidence without disruptions from human observation [4, 5]. The obtained raw images are reservable for subsequent analysis[5, 6]. Additionally, sample images offer extra detection details, such as the specific date, time, and ambient conditions during imaging [7].

The use of camera traps for monitoring squamates (snakes and lizards) has expanded, particularly in capturing behaviour and habitat use [8]. While camera trap surveys offer increased possibilities [3], manual review and classification of the captured images is time-consuming [9]. Automated processing faces challenges, including intraclass variation, unpredictable poses, lighting variations, motion blurriness, and cluttered backgrounds [10-12] Additional complexities arise from natural camouflage effects, partially displayed body fragments, and issues with distant or close targets [10]. Despite these challenges, the vast amount of data generated requires automated methods for efficient species detection and identification. Camera traps, proven effective since the 1990s in estimating tiger populations in Nagarahole National Park [13], demonstrate versatility across diverse species and habitats, supporting behavioural and ecological studies.

These challenges are well-known to computer vision researchers. In response to these issues, research in computer vision and machine learning has underscored the importance of an automatic classification framework. Our study delved into the application of deep learning architectures for wildlife

species classification in biological studies. Additionally, we fine-tuned self-trained deep learning models and machine vision algorithms for a demanding camera trap image dataset. The goal is to establish a framework capable of identifying *monkey prosimian, antelope duiker, civet genet, leopard, rodent, bird, hog* by categorizing them within their respective groups from a sizable camera trap dataset. This paper's contributions include:

- EfficientNetV2 Application: Demonstrates EfficientNetV2 efficiently handle unstructured, real-world wildlife images, offering high accuracy in species classification with optimized computational performance.
- Fine-tuning Approach: Illustrates the adaptation of a pre-trained EfficientNetV2 model for specific tasks, enhancing performance even with limited datasets.
- Dataset Utilization: Uses a diverse dataset from Tai National Park in Côte d'Ivoire, including various species and blank images, to highlight regional wildlife complexity.
- Contribution to Conservation: Enhances species identification accuracy in camera trap images, aiding wildlife conservation by improving biodiversity monitoring and population tracking.

The rest of the paper is organized as follows: Section 2 reviews existing camera trap image classification methods. Section 3 describes the transfer learning models used, fine-tuning of EfficientNetV2-L, data augmentation, pre-processing, and interpretability techniques like CAM [14]. Section 4 presents the dataset, experimental setup, and model evaluation using metrics such as accuracy and F1-score, with visual explanations. Section 5 summarizes findings and suggests future research directions.

2. Related work

Advancements in machine learning and computer vision have revolutionized the field of wildlife species classification in camera trap imagery. An examination of recent literature indicates that significant research has been undertaken on automated species identification through the application of machine learning techniques, with a predominant focus on mammals and birds [15]. Researchers conducted a series of snake identification experiments using diverse methodologies and pretrained models. Patel et al. [16] used real-time object detection and image classification with ResNet achieving the best accuracy in identifying nine snake species from the Galápagos Islands. Rajabizadeh and Rezghi [17] compared traditional methods with MobileNetV2, finding the latter to outperform. Abayaratne et al. [18] achieved a 90.5% accuracy in classifying six snake species from Sri Lanka using MobileNet. Progga et al. [19] optimized pretrained models, with the SGD optimizer and fivefold cross-validation producing the best results. The SnakeCLEF challenge introduced by SnakeCLEF provided labelled data for automatic snake species recognition experiments [20]. In 2020, Bloch et al. [21] used Mask R-CNN with EfficientNets, achieving a macro-averaging score of 0.594 in distinguishing 783 snake species. In the 2022 SnakeCLEF challenge, Yu et al. [22] investigated EfficientNets and transformer models, achieving a macro F1-score of 71.82%. In the same challenge, researchers in another study [23] achieved an improved score of 82.65% through an ensemble approach of pretrained and MetaFormer models. The authors in [24] automated the design of camera trap image classification networks for diverse edge devices in independent clusters using a regression tree-based neural architecture search. The proposed method took 6.5 hours to find a suitable network for the Jetson X2 edge device, demonstrating competitive accuracies in subsequent testing compared to both automatically and manually designed networks. In the study by [25], the performance of pre-trained ResNet-50 models with augmentation parameters was examined. The transfer learning approach using the ResNet-50 model showed promising accuracy, achieving 86% accuracy. These results emphasize the effectiveness of ResNet-50 in accurately classifying wildlife species from ecological camera trap images.

EfficientNet models have demonstrated superiority over other ConvNets in terms of both smaller size and better accuracy, particularly in the field of transfer learning. The authors of the study [26] emphasized that these models consistently reduce the number of parameters while surpassing the state-of-the-art accuracy of other models when evaluated on the ImageNet dataset. Notably, the lowest Top-1 accuracy score achieved by EfficientNet-B0 is 77.1%, which is significantly higher than ResNet-50's score of 76.0%. Building upon this study, a subsequent investigation by [27] introduced EfficientNetV2, a new family of convolutional networks that further enhances efficiency in terms of training speed and parameter usage compared to its EfficientNet predecessor. Moreover, on the ImageNet dataset, EfficientNetV2-L achieves a remarkable Top-1 accuracy of 85.7%, surpassing EfficientNet-B7, which achieves 84.7%. These results underscore the continuous

Table 1. The list of notations used in this paper.

Symbol	Description
$\mathbb{E}[\cdot]$	Expectation (expected value)
$\mathbb{P}(\cdot)$	Probability function
$\mathcal{L}$	Loss function
α	Learning rate
λ	Regularization parameter
μ	Mean of a distribution
σ	Standard deviation of a distribution
ρ	Correlation coefficient
γ	Discount factor in reinforcement learning
β	Momentum parameter in optimization
W	Weight matrix
Z	Intermediate layer output
H	Hidden layer activations
O	Output layer activations
L	Loss value
T	True labels (ground truth)
$\hat{y}$	Predicted output by the model
P	Probability distribution over classes
C	Number of classes
$p(x)$	Probability density function
$P(X = x)$	Probability mass function
$P(Y\|X)$	Conditional probability
$\mathrm{Var}(X)$	Variance of a random variable
$\mathrm{Cov}(X, Y)$	Covariance between variables
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution
$\mathcal{U}(a, b)$	Uniform distribution
$\mathcal{J}(\theta)$	Objective function
$\nabla_\theta \mathcal{J}(\theta)$	Gradient of the objective function

advancements within the EfficientNet series, with EfficientNetV2 emerging as the top-performing variant among the latest releases.

This study develops a wildlife species classification framework using Deep Convolutional Neural Networks (DCNNs) and evaluates three approaches: a self-trained CNN, ResNet-50, and EfficientNetV2. Instead of competing with existing state-of-the-art models, the focus was on exploring how fine-tuning and augmentation impact performance on a small dataset. The results reveal the advantages of robust networks and pretrained weights for feature extraction and classification, as well as the significance of augmentation parameters. These findings aim to advance ecological monitoring and improve wildlife species classification within the research community. Table 1 outlines the notation conventions used in this paper for vectors, matrices, random variables.

3. Materials and methods

3.1 Transfer learning models

To leverage the potential of transfer learning, we incorporated pre-trained deep neural network models into our experimental framework.

ResNet-50: ResNet-50 operates on the principle of skip connections, often referred to as residuals. These connections play a crucial role in alleviating the vanishing gradient problem and ensuring smoother gradient flow during training. The mapping function of ResNet-50 is learned by modelling the residual function. This mathematical representation is given by Equation 1.

$$H(x) = F(x) + x \tag{1}$$

where $H(x)$ signifies the learned mapping, $F(x)$ represents the residual function, and x denotes the input.

ResNet-50 is celebrated for its deep architecture, characterized by a multitude of residual blocks. It has gained widespread adoption across various computer vision tasks, particularly those involving complex image data.

EfficientNetV2-S: EfficientNetV2-S adopts a neural network scaling mechanism to prioritize computational efficiency while maintaining competitive accuracy. Its mapping function is constructed by Equation 2.

$$H(x) = MBConv_k\left(\sum_{i=1}^{N} SE_i\right)(x) \tag{2}$$

where $MBConv$ denotes Mobile Inverted Bottleneck Convolution [28], k is the kernel size of the intermediate depth-wise convolutional, and SEi represents squeeze-and-excitation blocks. This model incorporates efficient scaling strategies and squeeze-and-excitation blocks, making it particularly suitable for real-time image analysis where computational efficiency is paramount.

EfficientNetV2-M: Similar to EfficientNetV2-S, EfficientNetV2-M utilizes neural network scaling, but with increased capacity to strike a balance between efficiency and accuracy. The mapping function closely resembles that of EfficientNetV2-S, relying on $MBConv$ blocks and squeeze-and-excitation (SE) mechanisms. With a larger model size and the inclusion of SE blocks, EfficientNetV2-M offers versatility and is well-suited for a wide spectrum of tasks that demand a combination of efficiency and accuracy.

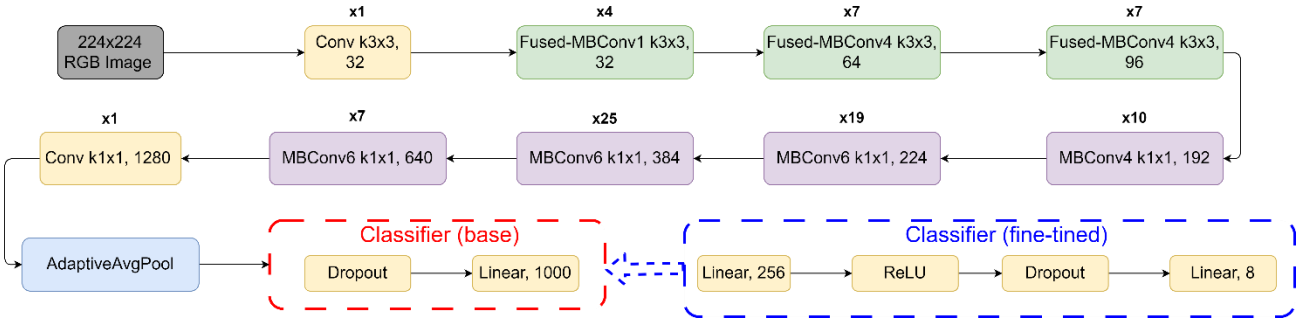

Figure. 1 The Base and Fine-tuned Architecture of EfficientNetV2-L

EfficientNetV2-L: EfficientNetV2-L, like its counterparts, employs neural network scaling but with even higher capacity, emphasizing high precision and accuracy. Its mapping function resembles that of EfficientNetV2-M, with inverted residual blocks and depth wise separable convolutions. As the largest and most computationally intensive model in the series, EfficientNetV2-L excels in precision-critical tasks where the highest levels of accuracy are essential, even at the cost of increased computational complexity.

These transfer learning models formed the cornerstone of our feature extraction and transfer learning experiments. Leveraging the knowledge encoded within extensive pre-trained datasets, we adapted these models to our specific classification task, achieving outstanding results.

3.2 Fine-tuning and network surgery

Fine-tuning in deep neural networks involves adjusting the weights and parameters of a pre-trained model for a specific task [29, 30]. It utilizes the knowledge and features learned during pre-training on a diverse dataset. This process transfers the learned knowledge from the original task to a related one by modifying the model's architecture, often by replacing or modifying the final layers for task-specific predictions. Fine-tuning can be made to the complete neural network or selectively to specific layers. In the latter scenario, the layers not undergoing fine-tuning are effectively "frozen" and remain unchanged during the backpropagation process. Additionally, a model can be enhanced by incorporating "adapters" with significantly fewer parameters than the original model. This allows for a parameter-efficient fine-tuning approach, focusing on tuning the weights of the adapters while keeping the remaining model weights frozen [31]. In this study, fine-tuning is performed on the EfficientNetV2-L model, which achieved the highest accuracy score in our experiment. Initially, we employ a scalpel to remove the last set of fully connected layers in a pre-trained neural network, specifically the "head" responsible for generating class label predictions (in red dotted rectangle). Subsequently, we substitute the excised head with a fresh set of fully connected layers, initialized randomly (in blue dotted rectangle). Figure. 1 provides a visual representation of our fine-tuned proposed architecture.

3.3 Augmentation

Data augmentation is a machine learning strategy employed to mitigate overfitting during the training of a model [32]. This approach involves training models on multiple variations of existing data, each slightly modified. Within the realm of computer vision, techniques for augmenting images have emerged as a notable implicit regularization strategy to mitigate overfitting in DCNNs. These techniques are widely utilized to enhance performance, as highlighted in previous works [33-35]. In this work, several image data augmentation techniques including zoom, shifting, Colour Jittering and Horizontal/Vertical Flipping operators are applied to the dataset.

Horizontal and Vertical Flipping is a method where images are mirrored to effectively address pose variations. This simple yet effective technique significantly increases the diversity of the training data, ultimately leading to improved model generalization. Inspiration for the use of horizontal and vertical flipping is drawn from prior studies that have successfully applied these techniques for image classification tasks in the field of computer vision. The authors in [36] demonstrated the effectiveness of data augmentation through image flipping in their seminal work on ImageNet classification with DCNNs. The authors in [37] further highlighted the utility of flipping transformations in improving image understanding tasks.

Colour Jittering involves introducing controlled variations in colour aspects such as brightness and contrast. This technique serves a dual purpose by enhancing the model's resilience to lighting

variations and improving its generalization capabilities. The choice of Colour Jittering as an augmentation technique is motivated by its potential to enhance the model's ability to generalize across different lighting conditions and environmental variations. The work of [38] and [39] demonstrated the efficacy of Colour Jittering in improving the robustness of deep learning models for image classification and recognition tasks.

The data augmentation process is illustrated in Figure. 2 and Figure. 3, showcasing the transformation of original images through horizontal and vertical flipping, as well as Colour Jittering. During the training phase, each batch of images undergoes random horizontal and vertical flipping. Additionally, Colour Jittering is applied to further enrich the training data. By introducing these variations, we aim to create a more diverse and extensive training dataset, enabling the model to generalize better on unseen test data.

3.4 Early stopping

In machine learning, early stopping serves as a regularization technique employed to prevent overfitting during the training of a learner using iterative methods like gradient descent. It monitors the model's performance during training and finding the optimal balance between complexity and generalization [40]. It tracks the validation loss, and the training process is stopped when the validation loss consistently increases, selecting the model with the lowest validation loss as the final model. These approaches iteratively enhance the learner to better align with the training data. Initially, this refinement improves the learner's efficacy on data beyond the training set. Nevertheless, there exists a threshold where refining the learner's fit to the training data leads to an escalation in generalization error. Early stopping rules guide the optimal number of iterations to prevent the learner from overfitting. The application of early stopping rules spans various machine learning methods, each with differing degrees of theoretical support [41], [42]. In this study, cross-entropy loss is utilized to track the improvement of validation loss. It is a widely used loss function in classification tasks, measuring the difference between predicted probabilities and true class labels. By minimizing the Cross-entropy loss during training, the model enhances its ability to make accurate class predictions. The Cross-entropy loss is calculated by Equation 3.

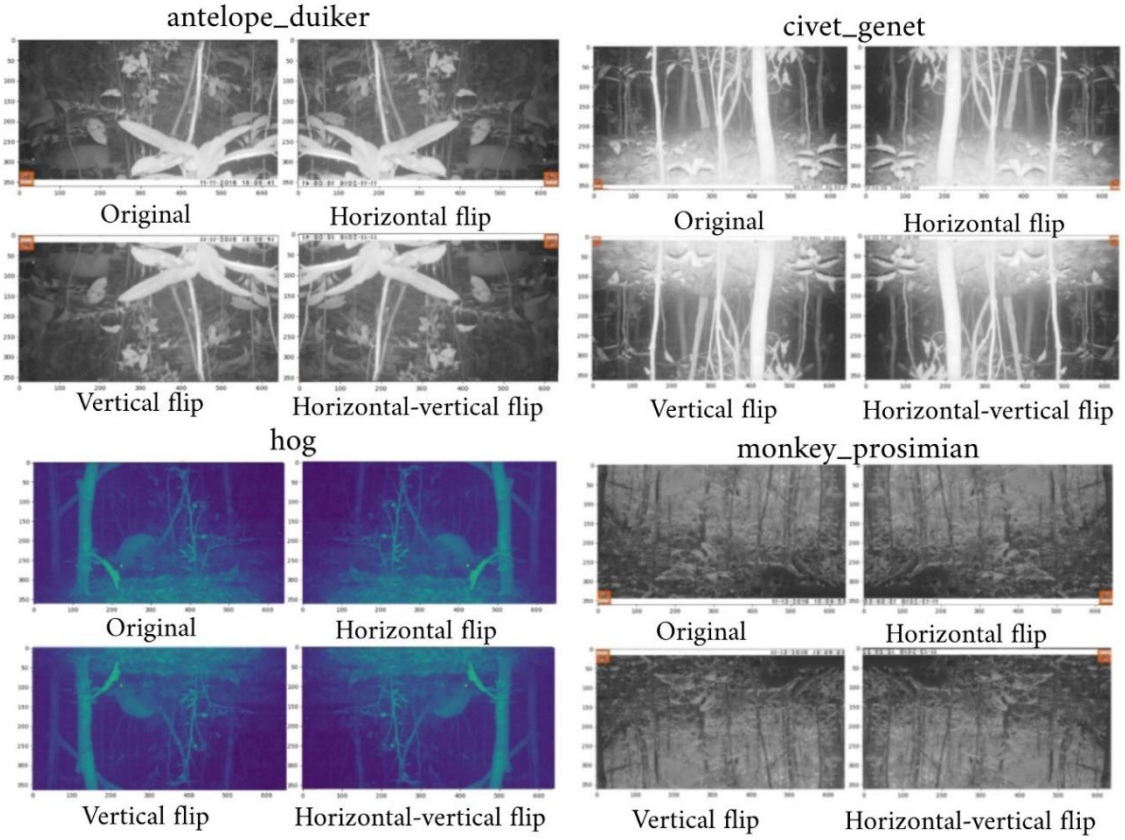

Figure. 2 Visual representation of Horizontal and Vertical Flipping results

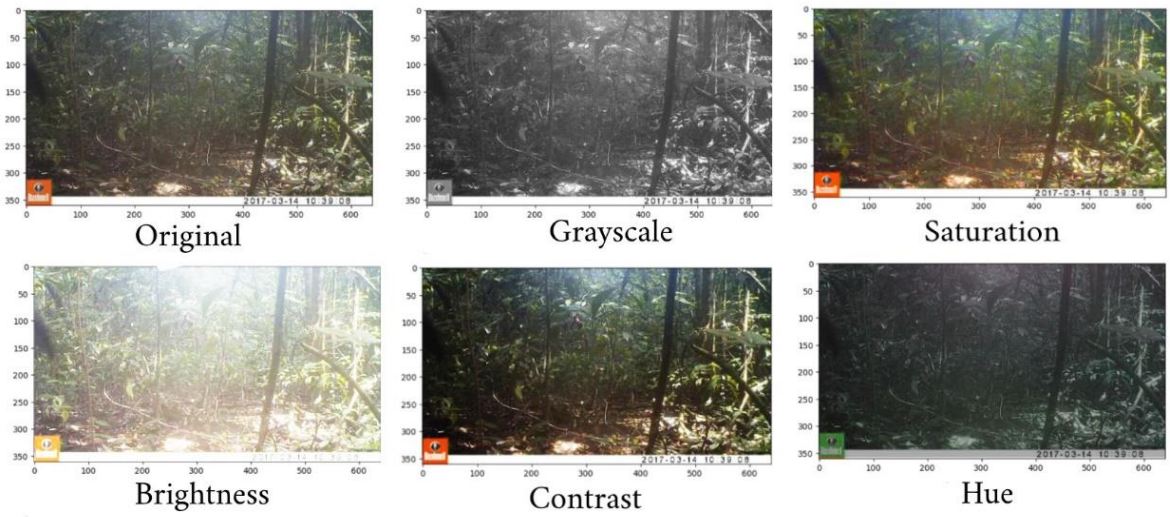

Figure. 3 Visual representation of Colour Jittering results

$$L_{CE} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log (p_{ij}) \quad (3)$$

Where:

- L_{CE} represents the Cross-entropy loss.
- N is the number of samples in the dataset.
- C is the number of classes.
- y_{ij} is an indicator function that equals 1 if the true class label for sample i is j, and 0 otherwise.
- p_{ij} is the predicted probability of sample i belonging to class j.

3.5 Visual explanation of deep learning models

Although deep learning has achieved remarkable accuracy in tasks like image classification, object recognition, and image segmentation, it faces a significant challenge in model interpretability. Understanding and debugging these models require a critical component called "interpretability" which is currently lacking in deep learning approaches. These models are often treated as "black boxes" making it difficult to grasp essential aspects such as where the network focuses its attention in an input image, which neurons were involved in making predictions, and the reasoning behind the final output.

To address this issue, Zhou et al. [14] have introduced a method known as Class Activation Mapping (CAM) for Convolutional Neural Networks with global average pooling. CAM aids deep learning researchers in debugging their network models by enabling them to locate objects in images without relying on bounding box annotations. By projecting class scores onto each image, CAM highlights the identifiable object regions that the CNNs have recognized. This visual information allows us to verify if the network is focusing and activating around the relevant patterns in the image, ensuring the model operates effectively.

To achieve successful species image categorization, it is crucial to carefully select the best CNN models used to extract image features. Therefore, this study evaluates CAM visualizations of various CNNs and their fine-tuning models for species classification tasks. Based on this evaluation, the most effective CNN models are chosen for the subsequent species image classification tasks. This approach aims to enhance model understanding and performance in the specific domain of species image analysis. Normally, the CAM process can be broken down into the following essential steps:

1) Feature Extraction: A pre-trained CNN extracts features from the input image through its convolutional layers.

2) CAM Generation: CAM calculates a weighted sum of feature maps for each class. It highlights regions in the feature maps that contribute most to a particular class's prediction.

3) Visualization: The CAM for a specific class is overlaid onto the input image. This highlights areas the model deems important for that class, offering insights into its decision-making process.

Figure. 4 shows the CAM results for different models on the task of species classification from camera trap images. The EfficientNetV2-L model is clearly superior to the other models in terms of its ability to identify and highlight species objects in the images. Its efficacy is further enhanced through fine-tuning process. The heatmaps generated by CAM for the fine-tuned EfficientNetV2-L model reveal that it consistently highlights visually appealing species

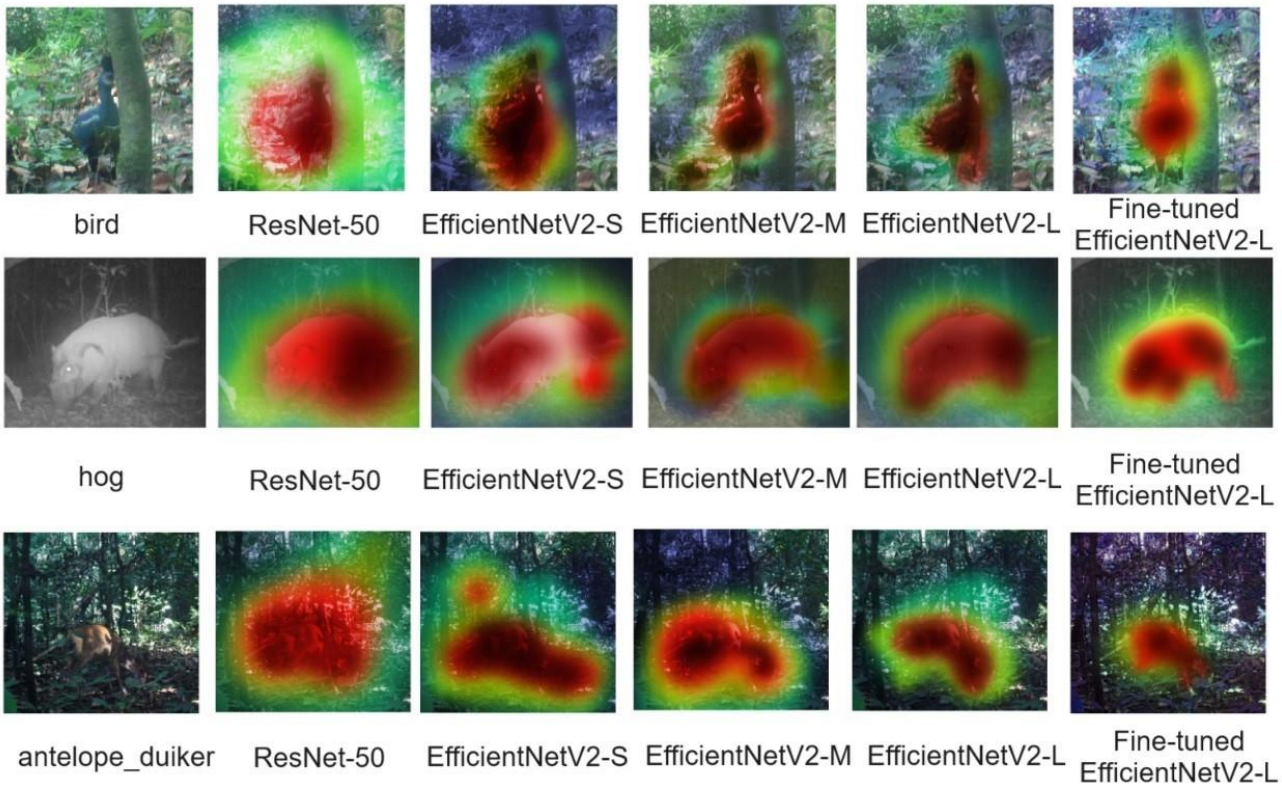

Figure. 4 Visual explanation of employed base models

objects in random images from the testing dataset (as depicted in the last column of Figure. 4). The model demonstrates attention to crucial areas of the species, including the head, body, and tail. In contrast, the CAM heatmaps for alternative models, particularly CNN models such as ResNet-50, EfficientNetV2-S, and EfficientNetV2-M, either indicate a dearth of object information or struggle to focus on pertinent sections of objects within the camera trap images.

This observation underscores the importance of CAM in assessing and interpreting model behaviour. CAM can be used to identify the regions of an image that are most important for a given classification prediction. This information can be used to understand how the model is making its predictions and to identify any potential biases in the model.

3.6 Confusion matrix

Within the domain of machine learning, particularly in the context of statistical classification, a confusion matrix, alternatively referred to as an error matrix [43], serves as a structured table layout facilitating the visualization of an algorithm's performance. This is typically applicable to supervised learning scenarios, while in unsupervised learning, it is commonly known as a matching matrix. In multi-class classification, the Confusion Matrix is a square matrix of size "N x N", where "N" represents the number of classes. Each row of the matrix corresponds to instances in an actual class, while each column corresponds to instances in a predicted class, or vice versa. Both variants are found in the literature [44]. The elements of the Confusion Matrix are as follows:

- True Positives (TP): The number of samples that are correctly predicted as positive for a particular class.
- True Negatives (TN): The number of samples that are correctly predicted as negative for a particular class.
- False Positives (FP): The number of samples that are incorrectly predicted as positive for a particular class.
- False Negatives (FN): The number of samples that are incorrectly predicted as negative for a particular class.

The Confusion Matrix allows us to calculate various performance metrics that can provide insights into the model's performance. Here are some commonly derived metrics:

1) Accuracy: serves as a measure of the overall correctness of the model's predictions. It is derived as the ratio of correctly classified samples to the total number of samples present in the dataset. This can be represented mathematically as Equation 4.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (4)$$

2) Precision: focuses on quantifying the proportion of true positive predictions among all the positive predictions made by the model. The precision score is computed using Equation 5.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{5}$$

In the realm of species detection, precision highlights the model's capability to correctly identify a specific species when it is indeed present.

3) Recall: gauges the proportion of actual positive samples that the model correctly identifies. The recall metric is calculated by Equation 6.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{6}$$

4) F1-score: serves as a harmonic mean between precision and recall and is instrumental in striking a balance between these two metrics. It yields a single value that takes into accounts both false positives and false negatives. The F1-score is computed by Equation 7:

$$\text{F1-score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \tag{7}$$

Notably, the F1-score is particularly valuable when dealing with imbalanced datasets, as it comprehensively considers both false positives and false negatives, thus offering a more holistic evaluation of the model's performance.

4. Results and discussions

4.1 Dataset

The dataset is provided for a challenge on DrivenData. The entire dataset originates from Tai National Park in Côte d'Ivoire and it is available at [45]. To ensure a comprehensive evaluation, the dataset was partitioned into two distinct sets: the training set and the testing set. The training set consists of 16,488 images, while the testing set contains 4,464 images.

The dataset encompasses eight distinct classes, each corresponding to a different wildlife species: *monkey prosimian, antelope duiker, civet genet, leopard, rodent, bird, hog, and blank* (representing images with no detected animals). Within the dataset, there may be a deliberate focus on certain species of particular interest, especially those that are of conservation concern or hold specific ecological significance. Consequently, more samples of these species might be included to ensure an adequate amount of data for accurate species recognition. Table 2 provides the count of images available in each class, offering insights into the distribution of the dataset across the different species categories.

Table 2. Distribution of images in each species class

Species	Number of Images
monkey_prosimian	2492
antelope_duiker	2474
civet_genet	2423
leopard	2254
blank	2213
rodent	2013
bird	1641
hog	978

A closer look at the structure of this dataset is presented in Table 3. It displays initial rows from the training CSV file, showcasing image labels along with their corresponding species classes. Each image in the dataset is assigned a specific identification (referred to as "id"), and if the value in the class field is 1, it indicates that the image belongs to that class; conversely, a value of 0 indicates that it does not belong to that class. The sum of values in each row will be 1, as the classes are mutually exclusive, meaning that each image in the dataset belongs to only one species class with no instances of multiple classes in a single image. Figure. 5 provides a closer look at what the images actually look like.

4.2 Experimental setup and training

The process in our experiment includes acquiring the dataset, setting up the environment, pre-processing data (which involves splitting it into training and evaluation sets and applying augmentation), fine-tuning the models, training the models, conducting evaluations, and finally, visualizing the efficiency of the models. All experiments were evaluated on afigure machine with Ubuntu 22.04 kernel 5.19, CPU Intel Core i7, and GPU NVIDIA GeForce RTX 2080 Ti. The deep learning framework used for all program implementation was PyTorch version 1.11, running on Python 3.9. All of the input images were scaled to the standard size that each network model accepts. In this study, images were set to 224x224 pixels for ResNet-50, EfficientNetV2-S, EfficientNetV2-M, EfficientNetV2-L.

ImageNet dataset was utilized to fine-tune the CNN models, enabling them to identify and categorize items within the datasets. The ImageNet dataset comprises over 1.2 million images and encompasses 1000 distinct classifications. Consequently, the final fully connected (FC) layers of all models, originally designed with 1000 outputs,

Table 3. CSV file samples with image labels and corresponding species classes.

id	antelope_duiker	bird	blank	civet_genet	hog	leopard	monkey_prosimian	rodent
ZJ005659	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ZJ007899	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
ZJ003690	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
ZJ003065	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
ZJ014753	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

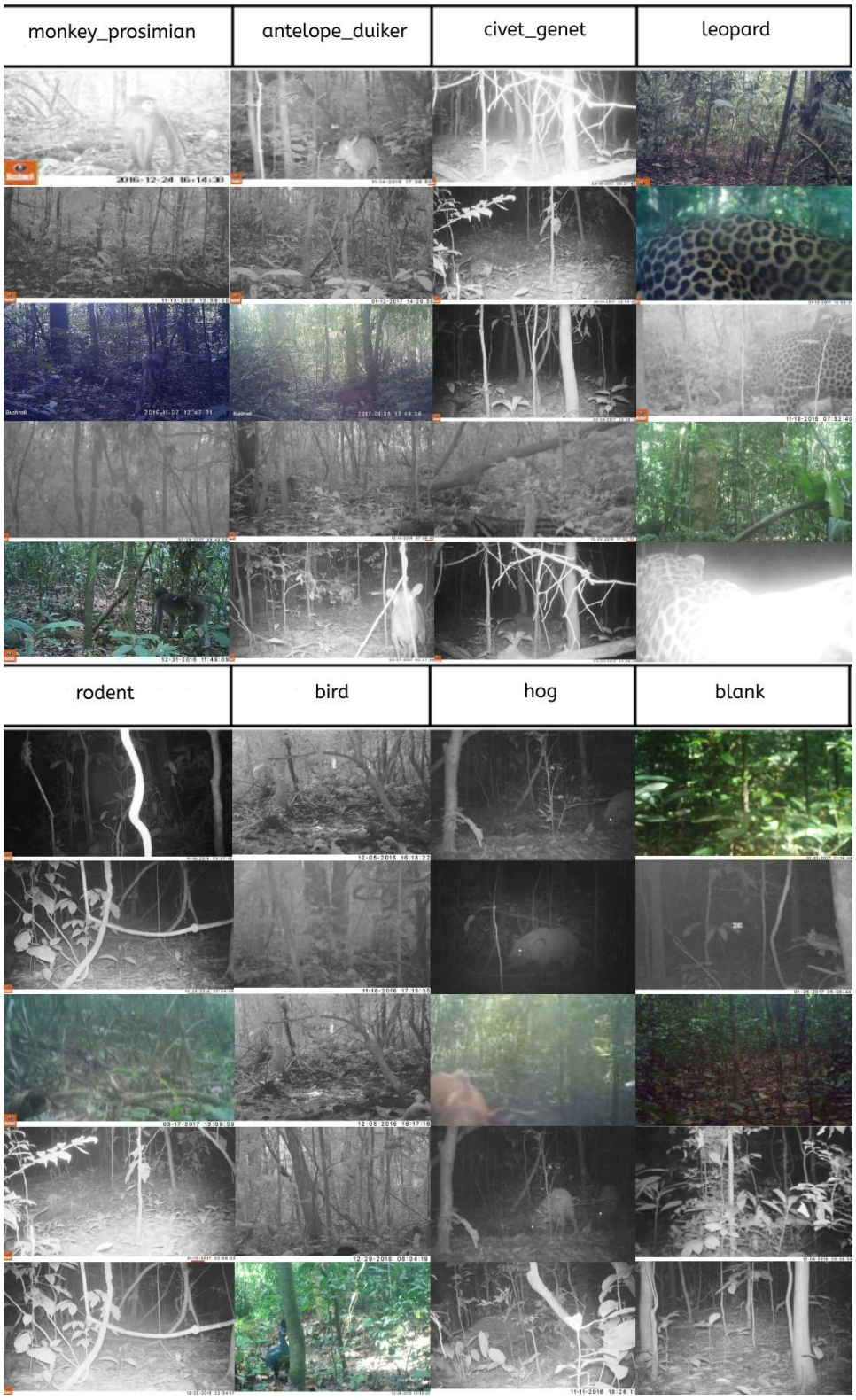

Figure. 5 Image samples from the dataset

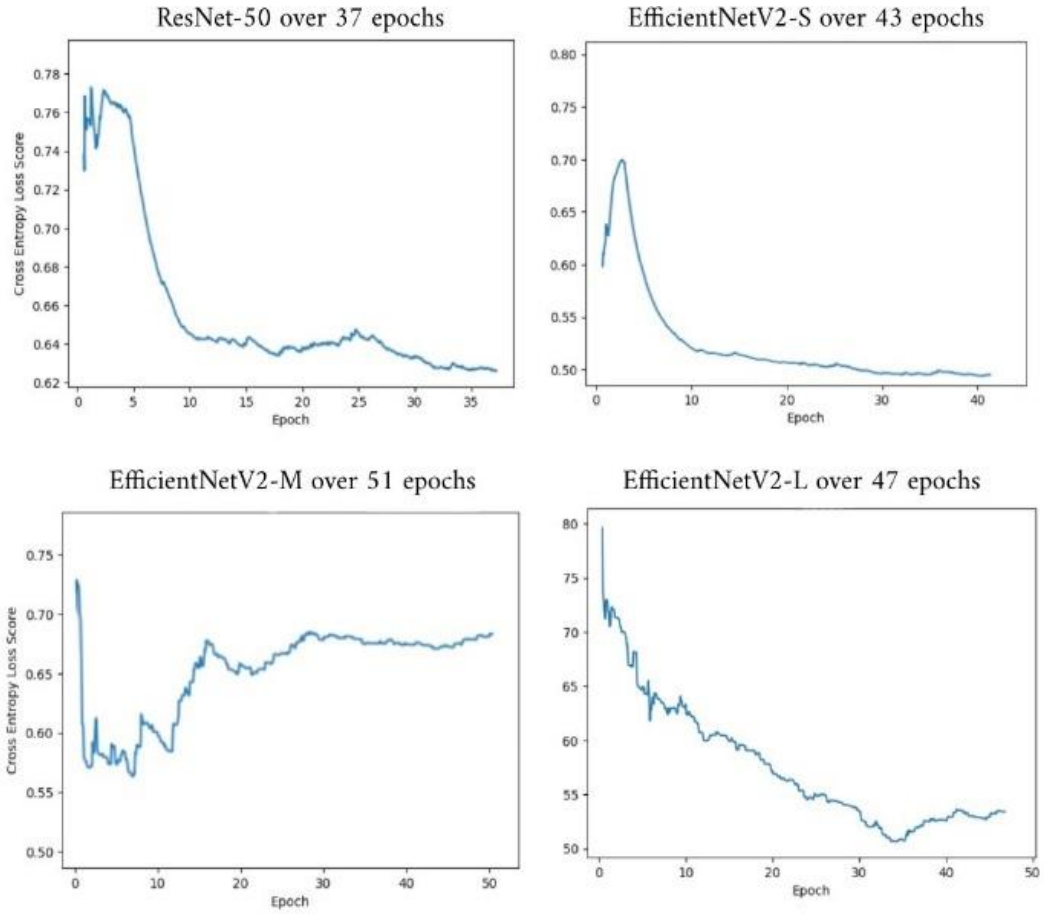

Figure. 6 Variation in loss score on base models

were adjusted to have 8 outputs in alignment with the 8 classes present in the dataset. To enhance efficiency, an early-stop technique was implemented during training, ceasing the process if the validation accuracy did not improve after three epochs. The parameters of successful models were preserved for subsequent testing at the conclusion of the training process. To facilitate a more comprehensive understanding and comparison of their feature extraction performance, the number of FC layers in all models was consistently maintained.

4.3 Results and discussion

In the training process, the early stopping technique was used with a minimal loss change set at 1e-3. As a result of employing this technique, there was no specific maximum epoch defined for model training. After training for a certain period, the base models (ResNet-50, EfficientNetV2-S, EfficientNetV2-M, and EfficientNetV2-L) ceased training when their cross-entropy loss scores showed no further improvement at the 37th, 43rd, 51st, and 47th epochs, respectively. Figure. 6 shows a concise overview of the loss scores across the epochs for each of these models. In general, the loss scores exhibited a gradual descent over time, eventually reaching very low values. This trend indicates significant improvement in the models' training performance and overall learning capabilities.

Evaluation results for the trained models are presented in Figure. 7. All metrics are above 82%, indicating that the four employed models performed well in the task of classifying species. Although the disparity between the base models' results in this study is relatively small, the EfficientNetV2-L outperformed its counterparts with accuracy, precision, recall, and F1-scores of 87.432%, 84.761%, 85.955%, and 88.501%, respectively. Table 4 and Table 5 compares our work to other classification methods.

However, it is important to note that the EfficientNetV2-L model was not trained on the specific dataset used in this study and may not perform optimally. To address this issue, the EfficientNetV2-L model was fine-tuned to explore the impact of fine-tuning methods and contribute to the effect of fine-tuned architecture. The architecture of fine-tuned EfficientNetV2-L model is described in Figure. 1.

Table 4. Accuracy, Precision, Recall, and F1-score of models in percent (%)

Models	Epochs	Batch Size	Accuracy	Precision	Recall	F1-score
ResNet-50	37	32	85.394	82.601	83.192	85.396
EfficientNetV2-S	43	32	86.257	83.439	84.141	86.288
EfficientNetV2-M	51	32	86.975	84.130	84.405	87.015
EfficientNetV2-L	47	32	87.432	84.761	85.955	88.501
Fine-tuned EfficientNetV2-L	42	32	88.822	86.941	87.638	87.193

Table 5. A comparison of our work to other methods

Method/Study	Model Used	Dataset	Accuracy (%)	Key Advantages
Our approach	Fine-tuned EfficientNetV2-L	Tai National Park, Côte d'Ivoire (8 species)	88.822	Superior accuracy; effective for complex, unstructured images; advanced data augmentation; conservation impact.
Patel et al. (2020) [16]	ResNet	Galápagos Islands (9 snake species)	86.0	Real-time object detection; effective for specific snake species.
Rajabizadeh & Rezghi (2021) [17]	MobileNetV2	General snake species (multiple regions)	93.16	Outperforms traditional methods; efficient for low-resource environments.
Abayaratne et al. (2021) [18]	MobileNet	Sri Lanka (6 snake species)	90.5	High accuracy for a small set of snake species.
Bloch et al. (2020) [21]	Mask R-CNN + EfficientNet	SnakeCLEF dataset (783 species)	59.4 (F1-score)	Integrates image and location data; lower precision on complex datasets.
Yu et al. (2022) [22]	EfficientNet + Transformers	SnakeCLEF 2022 dataset	71.82 (F1-score)	Combination of EfficientNet and transformers; effective for large-scale datasets.
Progga et al. (2021) [19]	Pretrained models + SGD	Snake species (2 classes)	91.30	A CNN based model
Jia et al. (2022) [24]	Custom-designed CNN	Camera trap images (various animal species)	97.38	Tailored for edge devices like Jetson X2; efficient for resource-constrained environments.
Islam et al. (2023) [25]	ResNet-50	Ecological camera trap images (multiple species)	86.0	Good baseline accuracy; effective for general wildlife classification

The fine-tuned EfficientNetV2-L model improved its performance on all classes (except the 'hog' class), outperforming the base model with an accuracy of 88.822% (an increasing of 1.39%), a precision of 86.941% (an increasing of 2.18%), a recall of 87.638% (an increasing of 1.68%), and an F1-score of 87.193%. As shown in Figure. 8 and Figure. 9, the fine-tuned model also improved its ability to distinguish between similar classes. This improvement is evident in a lower false positive rate for most classes and a higher true positive rate. These findings suggest that the fine-tuned model has a better understanding of the distinctive features that separate these two classes.

The high performance of these models, particularly in identifying and classifying wildlife species, suggests significant potential for improving ecological monitoring practices. The study emphasizes the utility of fine-tuned models and appropriate augmentation techniques for small datasets commonly encountered in wildlife research.

Overall, the study's experimental results underscore the efficacy of using advanced CNN models for wildlife species classification, with EfficientNetV2-L emerging as the top performer in this specific application. These findings contribute valuable insights for future research and practical implementations in ecological monitoring and and conservation efforts.

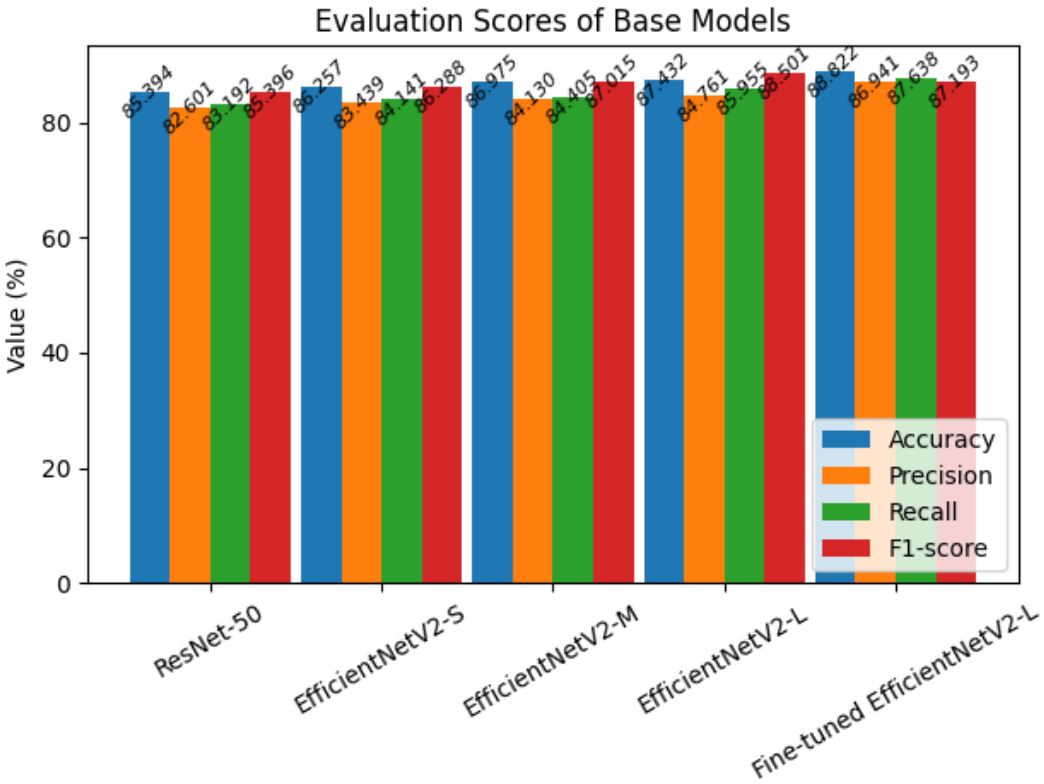

Figure. 7 Accuracy, Precision, Recall and F1-score of base models

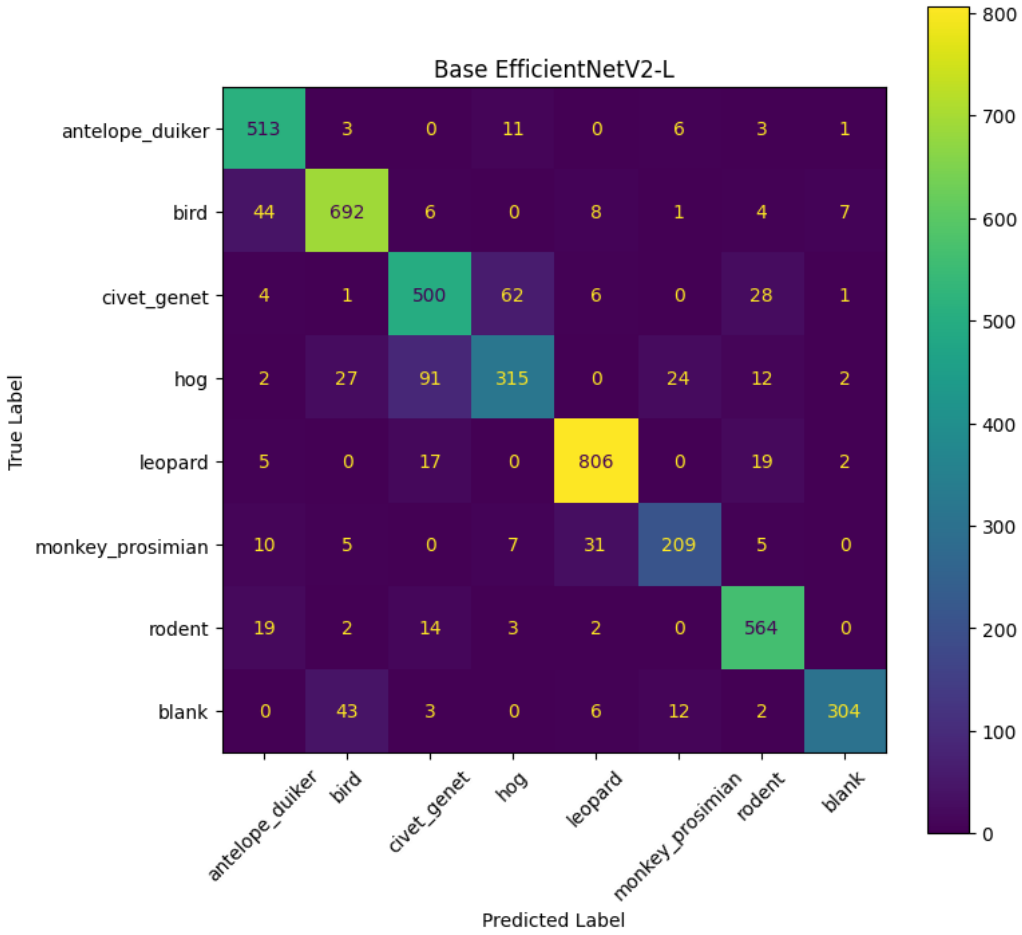

Figure. 8 Classification performance of EfficientNetV2-L model

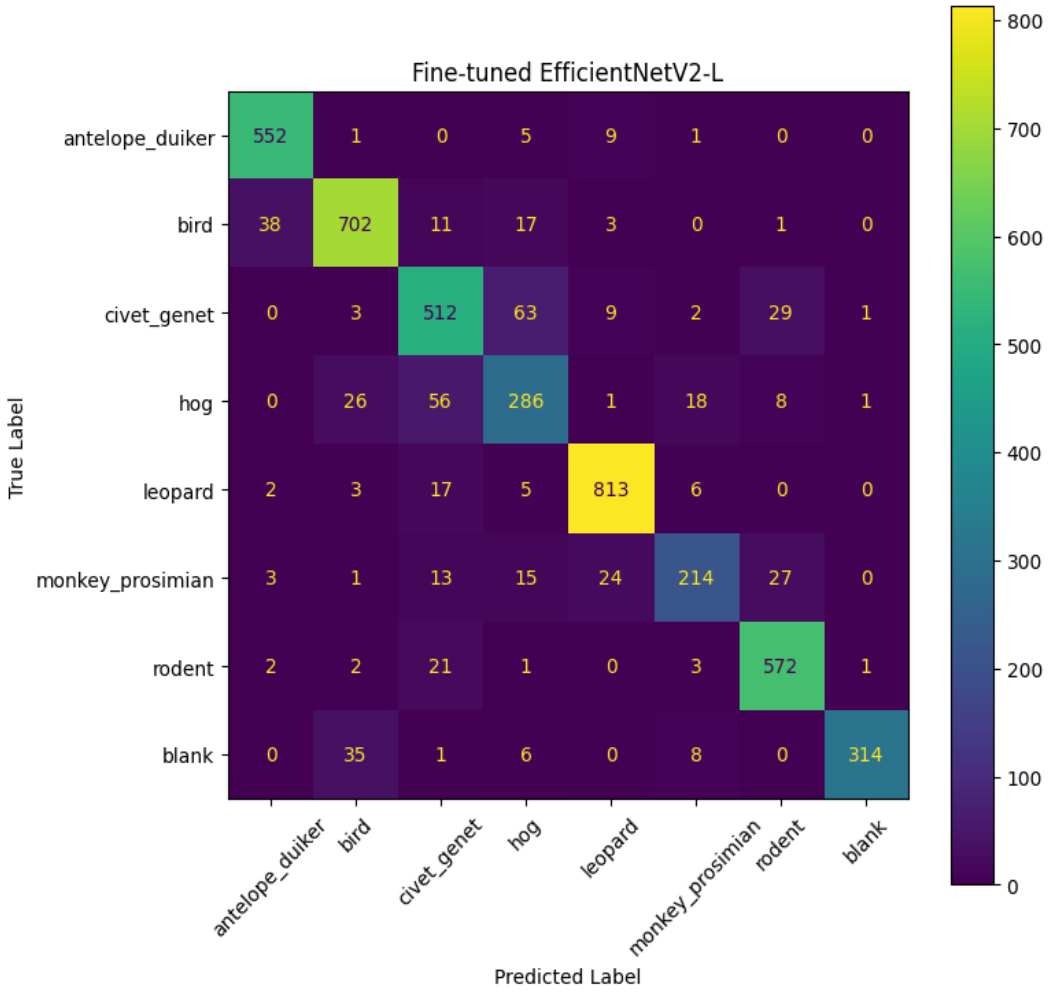

Figure. 9 Classification performance of fine-tuned EfficientNetV2-L model

5. Conclusions and future works

This study highlights the success of using transfer learning and fine-tuning DCNNs for wildlife species classification from camera trap images. Fine-tuning the EfficientNetV2-L model achieved an accuracy of 88.822%, precision of 86.941%, recall of 87.638%, and an F1-score of 87.193%, outperforming baseline models like ResNet-50. The theoretical foundation of our approach is based on transfer learning, which allows models to leverage features from large-scale datasets and adapt them to specialized tasks. These results underscore the practical utility of these techniques in improving ecological monitoring and supporting conservation efforts. Future research could investigate Vision Transformer [46] and YOLOV8 [47] to further improve classification accuracy and efficiency.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Thanh-Nghi Doan and Duc-Ngoc Le-Thi co-designed the study. Thanh-Nghi implemented the methodology and software, while Duc-Ngoc handled data curation and analysis.

Acknowledgments

This study was supported by the National Geographic Society, Microsoft AI for Earth, and the staff from An Giang University, Vietnam National University in Ho Chi Minh City, Vietnam.

References

[1] S. Leorna and T. Brinkman, "Human vs. machine: Detecting wildlife in camera trap images", *Ecological Informatics*, Vol. 72, p. 101876, 2022.

[2] P. K. Priya, T. Vaishnavi, N. Selvakumar, G. R. Kalyan and A. Reethika, "An Enhanced Animal Species Classification and Prediction Engine

using CNN", In: *Proc. of 2023 2nd International Conference on Edge Computing and Applications (ICECAA)*, 2023.

[3] A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith and C. Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna", *Scientific Data*, Vol. 2, p. 150026, 2015.

[4] Z. He, R. W. Kays, Z. Zhang, G. Ning, C. Huang, T. X. Han, J. J. Millspaugh, T. D. Forrester and W. J. McShea, "Visual Informatics Tools for Supporting Large-Scale Collaborative Wildlife Monitoring with Citizen Scientists", *IEEE Circuits and Systems Magazine*, Vol. 16, pp. 73-86, 2016.

[5] R. W. Kays, S. Tilak, B. Kranstauber, P. A. M. Jansen, C. Carbone, M. J. Rowcliffe, T. Fountain, J. Eggert and Z. He, "Monitoring wild animal communities with arrays of motion sensitive camera traps", *ArXiv*, Vol. abs/1009.5718, 2010.

[6] S. B. Islam, D. Valles and M. R. J. Forstner, "Herpetofauna Species Classification from Images with Deep Neural Network", *2020 Intermountain Engineering, Technology and Computing (IETC)*, pp. 1-6, 2020.

[7] C. S. Adams, W. A. Ryberg, T. J. Hibbitts, B. L. Pierce, J. B. Pierce and D. C. Rudolph, "Evaluating effectiveness and cost of time-lapse triggered camera trapping techniques to detect terrestrial squamate diversity", *Herpetological review*, Vol. 48, pp. 44-48, 2017.

[8] D. J. Welbourne, D. J. Paull, A. W. Claridge and F. Ford, "A frontier in the use of camera traps: surveying terrestrial squamate assemblages", *Remote Sensing in Ecology and Conservation*, Vol. 3, 2017.

[9] H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie and D. Q. Phung, "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring", In: *Proc. of 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 40-49, 2017.

[10] X. Yu, J. Wang, R. W. Kays, P. A. M. Jansen, T. Wang and T. S. Huang, "Automated identification of animal species in camera trap images", *EURASIP Journal on Image and Video Processing*, Vol. 2013, pp. 1-10, 2013.

[11] S. Schneider, G. W. Taylor and S. C. Kremer, "Deep Learning Object Detection Methods for Ecological Camera Trap Data", In: *Proc. of 2018 15th Conference on Computer and Robot Vision (CRV)*, pp. 321-328, 2018.

[12] M. S. Norouzzadeh, A. M. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning", In: *Proc. of the National Academy of Sciences of the United States of America*, Vol. 115, pp. E5716 - E5725, 2017.

[13] K. U. Karanth, "Estimating tiger Panthera tigris populations from camera-trap data using capture-recapture models", *Biological Conservation*, Vol. 71, pp. 333-338, 1995.

[14] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization", In: *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2016-Decem, p. 2921–2929, 2016.

[15] A. C. Burton, E. Neilson, D. Moreira, A. Ladle, R. Steenweg, J. T. Fisher, E. Bayne and S. Boutin, "Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes", *Journal of Applied Ecology*, Vol. 52, p. 675–685, 2015.

[16] A. Patel, L. Cheung, N. Khatod, I. Matijosaitiene, A. Arteaga and J. W. Gilkey, "Revealing the Unknown: Real-Time Recognition of Galápagos Snake Species Using Deep Learning", *Animals*, Vol. 10, 2020.

[17] M. Rajabizadeh and M. Rezghi, "A comparative study on image-based snake identification using machine learning", *Scientific Reports*, Vol. 11, 2021.

[18] S. Abayaratne, W. M. K. S Ilmini and T. G. I. Fernando, "Identification of Snake Species in Sri Lanka Using Convolutional Neural Networks", *Sri Lanka Association for Artificial Intelligence (SLAAI)*, 2021.

[19] N. I. Progga, N. Rezoana, M. S. Hossain, R. ul Islam and K. Andersson, "A CNN Based Model for Venomous and Non-venomous Snake Classification", In: *Analogical and Inductive Inference*, 2021.

[20] L. Picek, I. Bolon, A. M. Durso and R. L. R. de Castaneda, "Overview of the SnakeCLEF 2020: AutomaticSnake Species Identification Challenge", In: *Proc. of Conference and Labs of the Evaluation Forum*, 2020.

[21] L. Bloch, A. Boketta, C. Keibel, E. Mense, A. Michailutschenko, O. Pelka, J. Rückert, L. Willemeit and C. Friedrich, "Combination of Image and Location Information for Snake Species Identification using Object Detection and EfficientNets", In: *Proc. of Conference and Labs of the Evaluation Forum*, 2020.

[22] J. Yu, H. Chang, Z. Cai, G. Xie, L. Zhang, K. Lu, S. Du, Z. Wei, Z. Liu, F. Gao and F. Shuang, "Efficient Model Integration for Snake Classification", In: *Proc. of Conference and Labs of the Evaluation Forum*, 2022.

[23] C. Zou, F. Xu, M. Wang, W. Li and Y. Cheng, "Solutions for Fine-grained and Long-tailed Snake Species Recognition in SnakeCLEF 2022", In: *Proc. of Conference and Labs of the Evaluation Forum*, 2022.

[24] L. Jia, Y. Tian and J. Zhang, "Identifying Animals in Camera Trap Images via Neural Architecture Search", *Computational Intelligence and Neuroscience*, Vol. 2022, 2022.

[25] S. Binta Islam, D. Valles, T. Hibbitts, W. Ryberg, D. Walkup and M. Forstner, "Animal Species Recognition with Deep Convolutional Neural Networks from Ecological Camera Trap Images", *Animals*, Vol. 13, p. 1526, 2023.

[26] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *arXiv:1905.11946*, 2019.

[27] M. Tan and Q. V. Le, *EfficientNetV2: Smaller Models and Faster Training*, 2021.

[28] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", In: *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520, 2018.

[29] J. Quinn, J. McEachen, M. Fullan, M. Gardner and M. Drummy, *Dive Into Deep Learning: Tools for Engagement*, Corwin Press, 2019.

[30] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Dive into Deep Learning", *Journal of the American College of Radiology : JACR*, 2020.

[31] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal and C. Raffel, "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning", *ArXiv*, Vol. abs/2205.05638, 2022.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *Journal of Big Data*, Vol. 6, 2019.

[33] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem", In: *Proc. of 2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pp. 117-122, 2018.

[34] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *Journal of Big Data*, Vol. 6, pp. 1-48, 2019.

[35] S. Yang, W.-T. Xiao, M. Zhang, S. Guo, J. Zhao and S. Furao, "Image Data Augmentation for Deep Learning: A Survey", *ArXiv*, Vol. abs/2204.08610, 2022.

[36] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *Neural Information Processing Systems*, Vol. 25, 2012.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", In: *Proc. of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248-255, 2009.

[38] E. D. Cubuk, B. Zoph, J. Shlens and Q. V. Le, *RandAugment: Practical automated data augmentation with a reduced search space*, 2019.

[39] R. Zhang, P. Isola and A. A. Efros, *Colorful Image Colorization*, 2016.

[40] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu and T. Liu, *Understanding and Improving Early Stopping for Learning with Noisy Labels*, 2021.

[41] Y. Yao, L. Rosasco and A. Caponnetto, "On Early Stopping in Gradient Descent Learning", *Constructive Approximation*, Vol. 26, pp. 289-315, 2007.

[42] G. Raskutti, M. J. Wainwright and B. Yu, "Early stopping for non-parametric regression: An optimal data-dependent stopping rule", In: *Proc. of 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1318-1325, 2011.

[43] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy", *Remote Sensing of Environment*, Vol. 62, pp. 77-89, 1997.

[44] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation", *Mach. Learn. Technol.*, Vol. 2, 2008.

[45] "Competition Image Classification Wildlife Conservation Data".

[46] B. N. Patro and V. S. Agneeswaran, *Efficiency 360: Efficient Vision Transformers*, 2023.

[47] J. Terven and D. Cordova-Esparza, *A Comprehensive Review of YOLO: From YOLOv1 and Beyond*, 2023.