AUTOMATED MULTI-SPECIES CLASSIFICATION USING WILDLIFE DATASETS BASED ON DEEP LEARNING ALGORITHMS

Article · December 2023

DOI: 10.62991/MMT1996359772

AUTOMATED MULTI-SPECIES CLASSIFICATION USING WILDLIFE DATASETS BASED ON DEEP LEARNING ALGORITHMS

Sandhya Sharma[1]*, Sangam Babu Neupane[1], Bishnu Prasad Gautam[2], Kazuhiko Sato[1]

[1]Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

[2]Kanazawa Gakuin University, 10 Suemachi, Kanazawa, Ishikawa 920-1392, Japan

Abstract

The classification and identification of wildlife for monitoring and conservation purposes have become increasingly important, particularly as environmental degradation becomes a growing concern. Several studies have employed manual methods that are time-consuming, erroneous, and laborious for classification on wildlife datasets. A fully automated classification system is required to address this issue. The use of deep learning for the automatic identification of wildlife has been suggested. However, studies that compare and validate the use of different models in real-world monitoring scenarios are lacking. In this study, we collected wildlife image datasets from the Animals with Attributes repository, and we evaluated the performance of two mainstream Convolutional Neural Network (CNN) architectures, EfficientNetB0 and VGG16, in multi-species classification and identification. We deployed a multi-species classification model, based on deep-learning techniques, that could effectively recognize 37 distinct species categories. Our analysis showed that the EfficientNetB0 model outperformed the VGG16 model overall. The model was trained on a diverse dataset of 185,111 images and tested on 3,131 images, achieving over 80% accuracy and a top-5 accuracy of more than 90%. The F1-score, precision, and recall values for each species category exceeded 0.90, indicating high accuracy in identifying individual species. The model's prediction performance was validated through experimentation and through the Gradio and Tkinter interfaces, which showed the model to be highly accurate and reliable in image classification. This model could be used in various wildlife monitoring and conservation applications and could make a significant contribution to the field of computer vision. The high accuracy of the model in identifying individual species categories and its reliability in managing a large number of species categorizations make it a valuable tool for conservationists and researchers who seek to monitor and protect wildlife species.

Keywords: *Wildlife monitoring, computer vision, transfer learning, accuracy, top-5 accuracy, Gradio interface, Tkinter interface*

1. INTRODUCTION

Automated multi-species classification is crucial to ecological research, wildlife monitoring, and biodiversity conservation. Understanding various species' distribution, abundance, and behavior relies heavily on identifying and categorizing animal species in image databases. Over the past two decades, the utilization of camera traps for ecological research has revolutionized wildlife research and conservation efforts [1]. Camera traps with extended battery life and ample data storage capacity are a cost-effective and user-friendly tool, which makes them ideal for fieldwork [2, 3]. With the growing prevalence of camera traps in the field, the volume of image data that is available for species identification and classification has significantly increased. For researchers and conservationists, however, the manual categorization of species in the collected image datasets is difficult because it is labor- and time-intensive and prone to error [4]. To overcome these challenges, we provide a novel approach for deep learning-based automatic multi-species image categorization in this paper.

Previous research has employed deep-learning algorithms for classifying species, although the majority of them [5, 6, 7] have concentrated on a single species or a small number of species. This restricts the use of these models in situations where a large number of species need to be categorized in the real world. To the best of our knowledge, there is a lack in the literature of a framework that can automatically classify images of a variety of species. This paper addresses this issue by presenting a novel method for automated multi-species image categorization that can work for many species and produce reliable findings. Our approach is designed to be scalable and flexible for diverse species and image datasets, and it is based on cutting-edge Convolutional Neural Networks (CNNs). This study has the potential to significantly advance the field of automated species classification by filling a research gap. Through the use of deep-learning algorithms, we create an automated method for classifying several species from images, which decreases the time and effort needed for manual species categorization while increasing accuracy [8, 9]. In our method, a deep-learning model is trained on a sizable collection of images, each of which has been labeled with the species it represents. The algorithm is trained to extract attributes from photographs, such as form, color, and texture, and to utilize these attributes to predict the species in new, unobserved images. By automating the procedure and producing precise, dependable findings, our method seeks to manage the challenges of manual species categorization.

Future studies in the fields of ecological research, animal monitoring, and biodiversity conservation could benefit greatly from the findings of this study. The automated multi-species image classification system created in this work has the potential to dramatically decrease the time and effort needed for manual species classification while also producing findings that are more accurate; thus it is an important tool for many studies. Additionally, this research opens the door for future advancements in the area of automated species categorization, which would eventually improve our knowledge of biodiversity and conservation activities. In this study, we employ automated deep-learning algorithms to classify various wildlife species, and we further evaluate the model's efficacy using a semi-automated application.

2. MATERIALS AND METHODS

Image datasets are crucial for accurately classifying wildlife species, and camera traps provide a wealth of such datasets. However, autonomous multi-species classification should not attempt to use these datasets alone, as doing so could lead to low accuracy. To ensure the reliability and usefulness of the datasets, we performed preprocessing steps such as data cleaning, splitting the data into training and testing sets, and applying augmentation techniques. To improve classification accuracy, we performed a feature extraction approach on the pre-trained EfficientNetB0 and VGG16 models using different hyperparameters for the learning rate and activation functions. We evaluated the different models using metrics and visualizing the testing datasets during experimentation, and we deployed the higher-performing model in the Gradio Interface and the Tkinter Graphical User Interface (GUI) application for multi-species image classification (Figure 1).

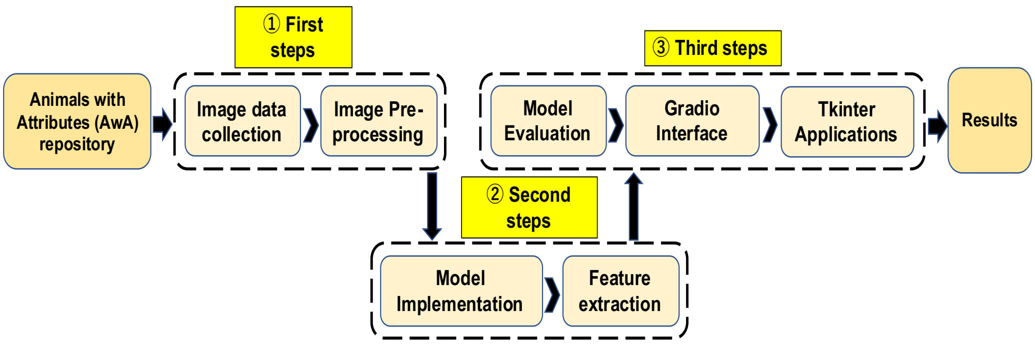


Fig. 1. Summarized research process

2.1. Image Data Collection

We retrieved 41,963 multi-class image datasets from the Animals with Attributes (AwA) repository for image classification [10]. These datasets belong to 50 different animal classes, including threatened species such as the Bengal tiger (*Panthera tigris tigris*), the giant panda (*Ailuropoda melanoleuca*), the humpback whale (*Megaptera novaeangliae*), and zebra (*Equus quagga*) (Figure 2). The AwA repository is publicly accessible. It also includes domesticated and wild animal image datasets and is used in computer vision and machine-learning research for precise species recognition. These image datasets provide a comprehensive and diverse representation of various animal species and can be used to train and evaluate image classification models. The collection of these datasets was performed in December 2022 and provides a valuable resource for researchers and developers working in the field of computer vision and image classification.

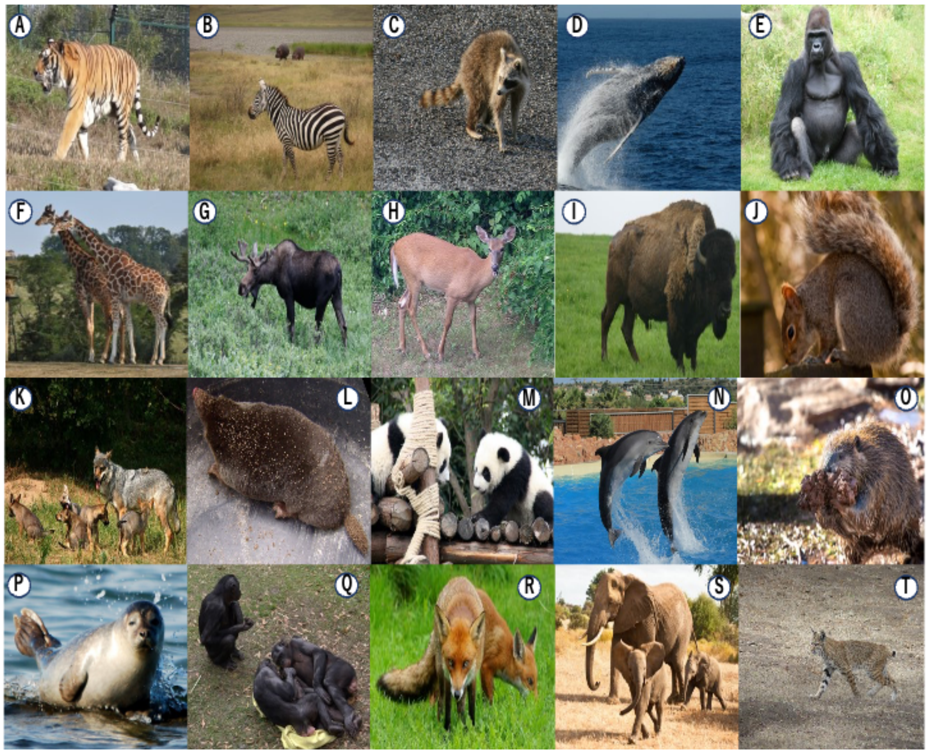


Fig. 2. Sample images of AwA datasets: A) Bengal tiger, *Panthera tigris tigris.* B) Plains zebra, *Equus quagga.* C) Raccoon, *Procyon lotor.* D) Blue whale, *Balaenoptera musculus.* E) Gorilla, *Gorilla spp.* F) Giraffe, *Giraffa tippelskirchi.* G) Moose, *Alces spp.* H) Deer, *Odocoileus spp.* I) American bison, *Bison bison.* J) Squirrel, *Sciurus spp.* K) Wolf, *Canis spp.* L) Mole, *Talpa spp.* M) Giant panda, *Ailuropoda melanoleuca.* N) Dolphin, *Delphinus spp.* O) Beaver, *Castor spp.* P) Seal, *Phoca.* Q) Chimpanzee, *Pan spp.* R) Fox, *Vulpes spp.* S) African elephant, *Loxodonta Africana.* T) Bobcat, *Lynxes spp*.

2.2. Image Preprocessing

After collecting image datasets for multi-species classification, we performed several preprocessing steps to improve their quality and usefulness. This included data cleaning to eliminate bias and irrelevant information, as well as splitting datasets into training and testing sets. Moreover, augmentation techniques were also used to balance the uneven distribution in the training datasets, which improved the accuracy and effectiveness of the image classification models. These steps are critical to ensuring that image datasets are reliable and can be used to support research and conservation efforts.

2.2.1. *Cleaning datasets*

During the initial filtering state, the datasets collected from the AwA repository were refined. The datasets included both domesticated and wild animals, but as our research was intended to conduct a multi-species wildlife classification analysis, only the image datasets of wild animals were used, and the domesticated image datasets were eliminated. This resulted in a total of 37 distinct classes of wild animals out of the initial 50 different classes. Additionally, some of the class datasets contained a mixture of domesticated and wild animals, so the domesticated images were considered to be outliers and were subsequently removed. This refinement process helped to eliminate any source of bias or irrelevant information that could have skewed the results.

2.2.2. *Split the datasets*

After the filtration, we obtained 31,144 image datasets of wildlife representing 37 distinct species. The datasets were then divided into a training set and a testing set, with the training set accounting for 90% of the samples (28,013) and the testing set accounting for the remaining 10% (3,131) that were used to train the machine-learning model and evaluate its performance, respectively. To ensure fairness, the datasets were randomly shuffled before the division was made (Figure 3).

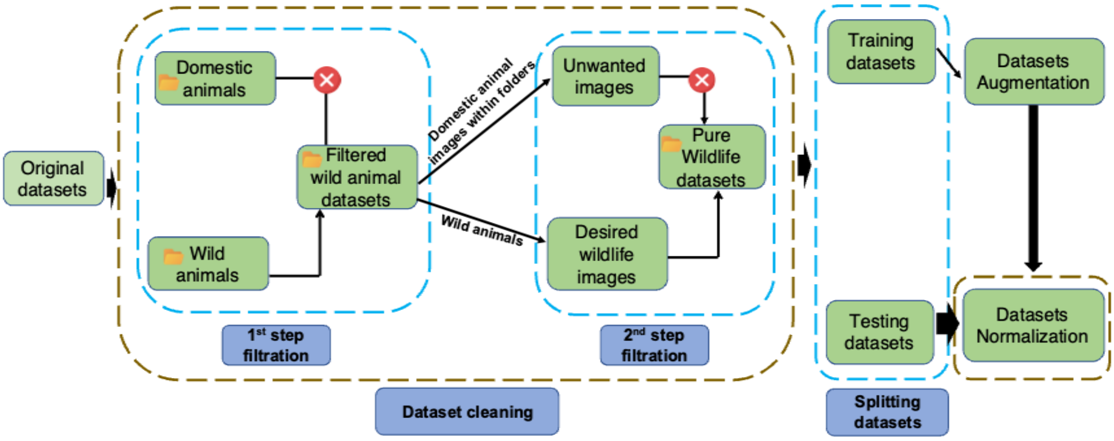


Fig. 3. Workflow for preprocessing image datasets

2.2.3. *Data augmentation*

We applied four types of data augmentation by slightly altering the original images. We rotated, zoomed, flipped, and shifted images both horizontally and vertically. Data augmentation was performed to balance the uneven distribution of image data between the different species in the training sets [11]. Some classes, like Mole, had very few images (N = 90), while others like the tiger had many more (N = 4,999). This imbalance could have led to biases in the evaluation of the machine-learning model [12, 13]. By augmenting the data, the classes with few images were increased until they had a number of images roughly equal to the number of images in the other classes (Mole; N = 5,013), providing a more balanced representation of the data for the model to learn from and preventing biases in the model predictions (Figure 4). For the model, we used 1,851,11 image datasets as training sets, whereas the number of images in the testing sets remained the same as the original testing dataset because data augmentation was performed only on the training sets. We then normalized the training and testing image datasets to improve the performance, stability, and generalization of the model by scaling the pixel value between 0 and 1.

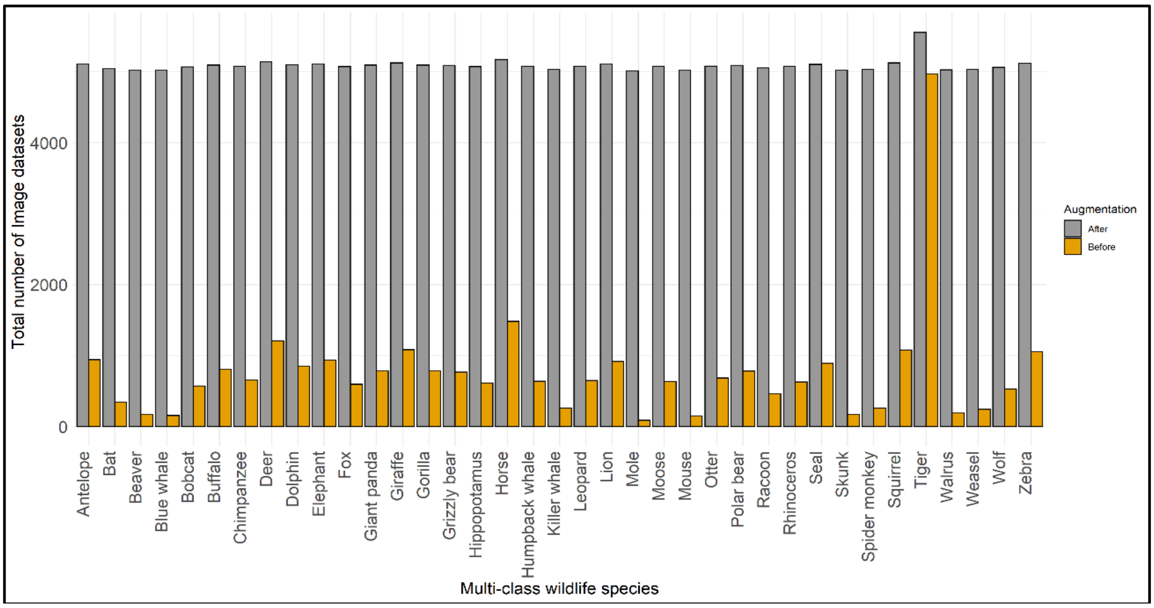


Fig. 4. Total number of image datasets in each class before and after the data augmentation

2.3. Model Implementation and Evaluation

In the realm of wildlife ecology, transfer learning is frequently employed to train machine-learning models when data are scarce [14]. This methodology capitalizes on the knowledge garnered from a pre-existing model, which leads to better accuracy even with small training datasets. In this experiment, we implemented transfer learning techniques using TensorFlow to classify images of wild animals with pre-trained models such as VGG16 [15] and EfficientNetB0 [16]. By leveraging prior knowledge of data patterns and features, we performed a feature extraction approach on the pre-trained models to enhance their performance (Figure 5). To ensure that the dense layers were tailored to the specific data requirements, the fully connected layers of all architectures were frozen. In our investigation, we utilized a constant dense layer with 5,423,397 trainable parameters. This approach allowed us to ensure consistency across all models and make fair comparisons of their performance. We employed Rectified Linear Unit (ReLU) and leaky ReLU as the activation functions, with learning rates of 0.0001, 0.001, and 0.01 to compare the performance of different models under different hyperparameters. We utilized batch normalization and dropout for the regularization of all the models during the experiment. Subsequently, we assessed the testing datasets using various performance metrics, including accuracy and top-5 accuracy, but recall, precision, and F1-score were determined only for the model with high accuracy (hereafter the best model). We conducted the training and evaluation of the architecture using the 12th Gen Intel® Core™ i7-1200F 2.10 GHz processor and 32.0 GB RAM. We processed 12 models with a constant batch size of 32.

2.4. Experimental Setup

The learning rate is a crucial hyperparameter that governs the magnitude of the updates made to a model's performance during training. In this study, we evaluated three different learning rates (0.0001, 0.001, and 0.01) across all the models to investigate the trade-off between convergence speed and overshooting or oscillation. We conducted experiments with various learning rates to identify the optimal value that would enable the model to make precise predictions on unseen test data. Additionally, we explored the use of different activation functions, including ReLU and leaky ReLU, to introduce non-linearity into the model, allowing it to learn more complex patterns and make more accurate predictions. To evaluate the performance of the models, we utilized several metrics, including accuracy score, top-5 accuracy, recall, precision, and F1-score. These metrics provide a comprehensive understanding of a model's real-world performance and are crucial in assessing its ability to predict unseen test images. Overall, the experimentation and evaluation of hyperparameters and performance metrics help improve the performance of machine-learning models in various applications.

2.4.1. Hyperparameter tuning using Learning rate

The learning rate is a critical hyperparameter that governs the magnitude of the updates made to a model's performance during the training process [17]. It acts as a constraint on the step size, ensuring that the model reaches the optimal solution without overshooting or oscillating around it. In this study, we evaluated three different learning rates, namely 0.0001, 0.001, and 0.01, across all of our models. A small learning rate like 0.0001 can impede convergence, but it can also prevent the model from overshooting the optimal solution. Conversely, a large learning rate like 0.01 can facilitate rapid convergence but may lead to overshooting the optimal solution or even diverging from the solution. A moderate learning rate like 0.001 can strike a balance between the two extremes, and it offers a reasonable convergence rate and avoids overshooting or oscillation [17]. However, it is crucial to select an appropriate learning rate based on the species and dataset in use. Therefore, we performed experiments with various learning rates to identify the optimal value that would enable our model to make precise predictions on unseen test data.

2.4.2. Hyperparameter tuning using activation functions

In our experimentation, we explored the use of different activation functions to improve the performance of our models. The two activation functions we utilized were ReLU and leaky ReLU. ReLU is a widely used activation function that returns the input value if it is positive and returns zero otherwise, which is computationally efficient and helps to mitigate the vanishing gradient problem. Leaky ReLU is a modified version of the ReLU activation function that returns the input value if it is positive and a small negative value otherwise [18]. The use of the activation function helps to introduce non-linearity into the model, allowing it to learn more complex patterns and make more accurate predictions.

2.5. Performance Metrics

In our experiment, we utilized several metrics to evaluate the performance of our models in predicting unseen test images. These metrics included accuracy score, top-5 accuracy, recall, precision, and F1-score. It is a widely used metric that provides a general idea of how well the model is performing on the test data [19]. Top-5 accuracy represents the proportion of test images that contain the correct label within the top five predicted classes. It is particularly useful when dealing with datasets that have a large number of classes. Recall, precision, and F1-score are measures that are used to evaluate the performance of the model. Recall represents the proportion of true positives (correctly identified positive) to the total number of actual positives; numerically, Recall = True Positive / (True Positive + False Negative). Precision represents the proportion of true positives to the total number of predicted positives; numerically, Precision = True Positive / (True Positive + False Positive). F1-score is the harmonic mean of recall and precision, and it provides a more balanced evaluation of the models' performance of both false positives and false negatives; numerically, F1-score = 2 × Precision × Recall / (Precision + Recall). Multiple model evaluation metrics provide a comprehensive understanding of the model performance in predicting unseen test images, which is crucial in assessing the model's real-world performance.

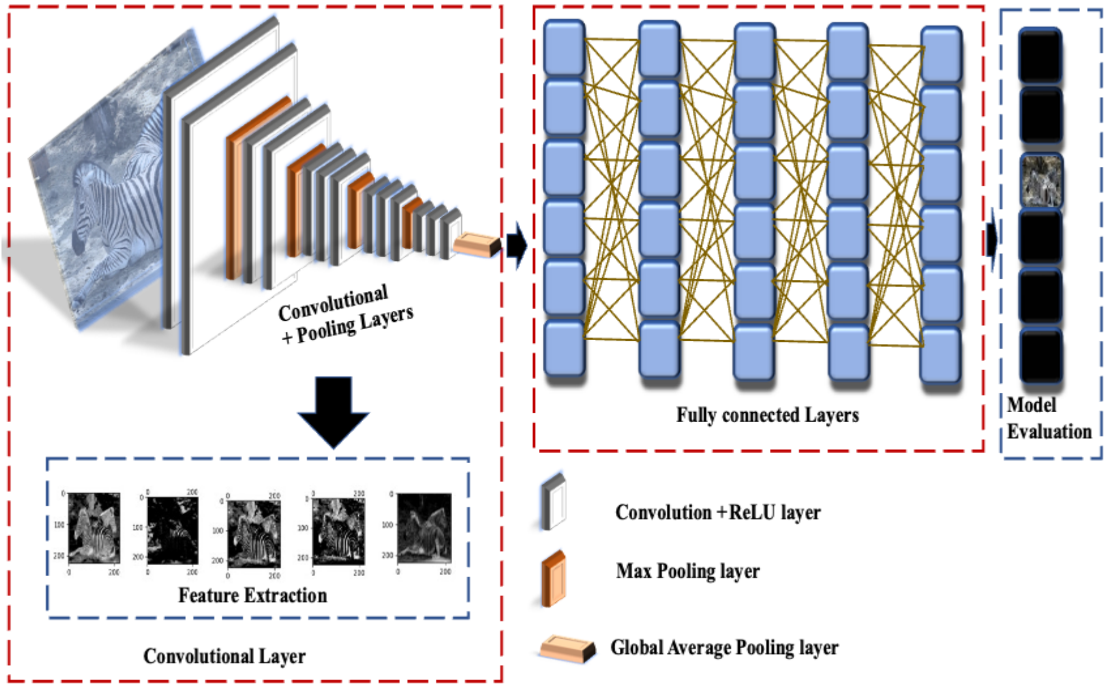


Fig. 5. Feature extraction of VGG16 architecture used for the study

2.6. Model Evaluation Through Applications Interface

We used a combination of the Gradio and Tkinter interfaces to evaluate and improve the performance of our best machine-learning model. By utilizing Gradio, we were able to visualize the model's predictions in real time and identify errors, which allowed us to improve its performance on previously unseen test images. Tkinter provided a user-friendly interface for evaluating and classifying new images, which made it easier to assess the model's performance and identify areas for improvement. Overall, the use of these tools helped to streamline the evaluation process, resulting in a more efficient and effective approach for improving the performance of our machine-learning model on previously unseen test images.

2.6.1. Gradio interface

During our experiment, we utilized Gradio to evaluate the performance of our model on unseen test images. Gradio is a Python library that provides a simple and intuitive interface for testing and evaluating machine-learning models [20]. We used Gradio to deploy our best trained model, which allowed us to input test images and obtain predictions from the model. We were able to visualize the model predictions, check for errors, and assess the model’s overall performance on the test data. By using Gradio during our experiment, we obtained real-time feedback on the model performance on unseen images, and we evaluated and improved our model in a fast and interactive way.

2.6.2. Tkinter interface

We utilized Tkinter, a GUI interface, to assess previously unseen testing datasets. The resulting interface, dubbed Multi-Species Image Classifier, featured five buttons. The first button, “Select Image Directory,” provided pathways for the importation of image datasets. The second button, “Select Destination Directory,” enabled the saving of our multi-species image datasets into several class name folders. The third button, “Classify Image,” classified the previously unseen image datasets automatically into several class names predicted by our best models. Unclassified image datasets were saved in the folder labeled “others.” The fourth button, “View Image,” facilitated viewing the testing images, and the fifth button, “Next Image,” allowed for changes to the images located in the selected image directory path. The images were classified and displaced through the interface.

3. RESULTS

3.1. *AwA Datasets*

After preprocessing the datasets through filtering and augmentation techniques, we selected a total of 185,111 images for the training dataset and 3,131 images for the testing dataset. These images belonged to 37 distinct species categories. The images used for training and testing were all standardized to have a resolution of 224×224 pixels (Table 1). This dataset provided a diverse set of images, representing multiple species, that was used to evaluate the performance of our model in identifying the correct species. Using this dataset, we were able to ensure that our model was trained on a representative sample of images and could generalize well to new, previously unseen images.

Table 1. Properties of image datasets used for model training and evaluation

Species categories	Training datasets after filtering		Testing datasets	Image resolution (pixels)
	Before augmentation	After augmentation		
37	28,013	185,111	3131	224×224

3.2. Experimental Results

3.2.1. Best model selection

In our study, we tested various hyperparameters for both the EfficientNetB0 and VGG16 models to identify multiple wildlife species. We found that the models performed well, achieving over 80% accuracy and a top-5 accuracy of greater than 90% (as shown in Table 2). However, our analysis showed that the EfficientNetB0 model outperformed the VGG16 model overall. After experimenting with different hyperparameters, we determined that the best configuration was to use EfficientNetB0 with ReLU activation functions and a learning rate of 0.0001. This configuration achieved the highest accuracy on the testing datasets, with 92% accuracy and a top-5 accuracy of 99% (Table 2), and was considered the best model configuration for our experiment. This highlights the model's effectiveness in accurately identifying species. The VGG16 model took considerable time (1.79 hours) when using ReLU activation functions and a learning rate of 0.01, whereas the EfficientNetB0 model took the least time (0.60 hours) when using the leaky ReLU activation function and a learning rate of 0.001. Our best model required 0.77 hours to train, which was found to be quite efficient compared with other models (Table 2).

3.2.2. Performance of best model

In our study, we examined the number of epochs and the accuracy of the best model, which was determined to be the EfficientNetB0 model with ReLU activation functions and a learning rate of 0.0001. We found that as the number of epochs increased, the accuracy of the model on both the training and testing datasets improved (Figure 6). Furthermore, we observed that the training and testing curves were closely aligned, indicating that the model did not overfit the training data. This alignment suggested that the model not only predicted the training data, but that it also generalized well to the unseen testing data.

Table 2. Overall accuracy score with the time to train the EfficientNetB0 and VGG16 models for different hyperparameter values. Here bold indicates the best model

Experiment number	Architecture	Learning rate	Activation function	Accuracy score (%)	Top-5 Accuracy (%)	Time (hours)
1	EfficientNetB0	0.0001	Leaky ReLU	881	92	0.79
2	**EfficientNetB0**	**0.0001**	**ReLU**	**92**	**99**	**0.77**
3	EfficientNetB0	0.001	Leaky ReLU	79	91	0.60
4	EfficientNetB0	0.001	ReLU	82	92	0.79
5	EfficientNetB0	0.01	Leaky ReLU	76	91	0.78
6	EfficientNetB0	0.01	ReLU	73	91	0.77
7	VGG16	0.0001	Leaky ReLU	89	98	1.73
8	VGG16	0.0001	ReLU	87	98	1.78
9	VGG16	0.001	Leaky ReLU	86	97	1.71
10	VGG16	0.001	ReLU	85	98	1.73
11	VGG16	0.01	Leaky ReLU	89	92	1.73
12	VGG16	0.01	ReLU	87	95	1.79

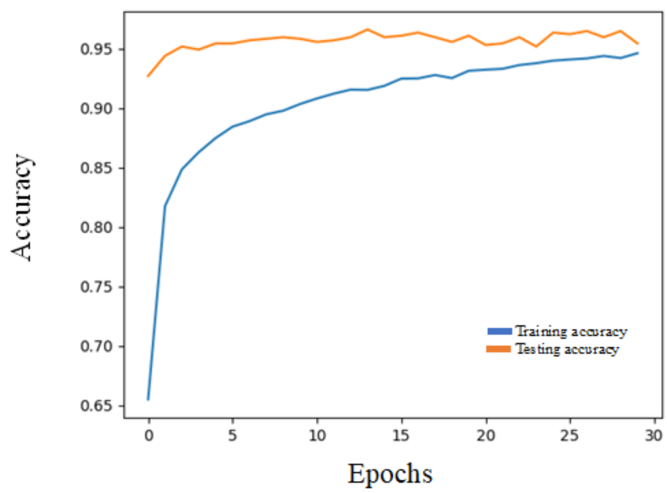


Fig. 6. Accuracy curves for training and testing, plotted from epoch 0 to epoch 30

In this study, we assessed the performance of our best model in terms of F1-score, precision, and recall for all multi-species categories. Our findings indicated that the F1-score, precision, and recall values for each species category exceeded 0.90, demonstrating high accuracy in identifying individual species categories (Figure 7). These results suggest that our best model excelled in recognizing different species, and was suitable for species classification tasks. The high F1-score, precision, and recall value achieved by our best model underscored its efficacy in accurately identifying various species categories.

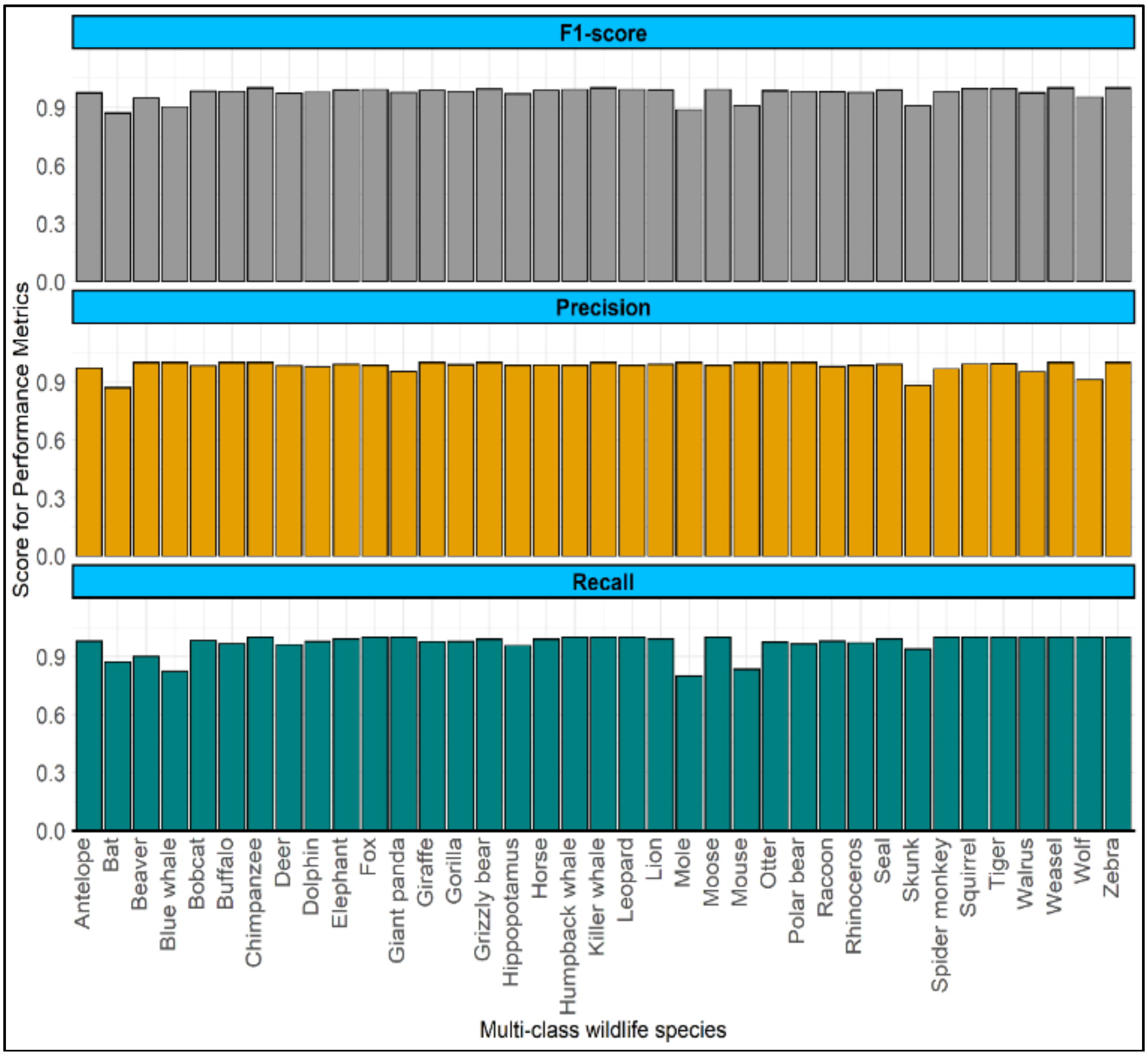
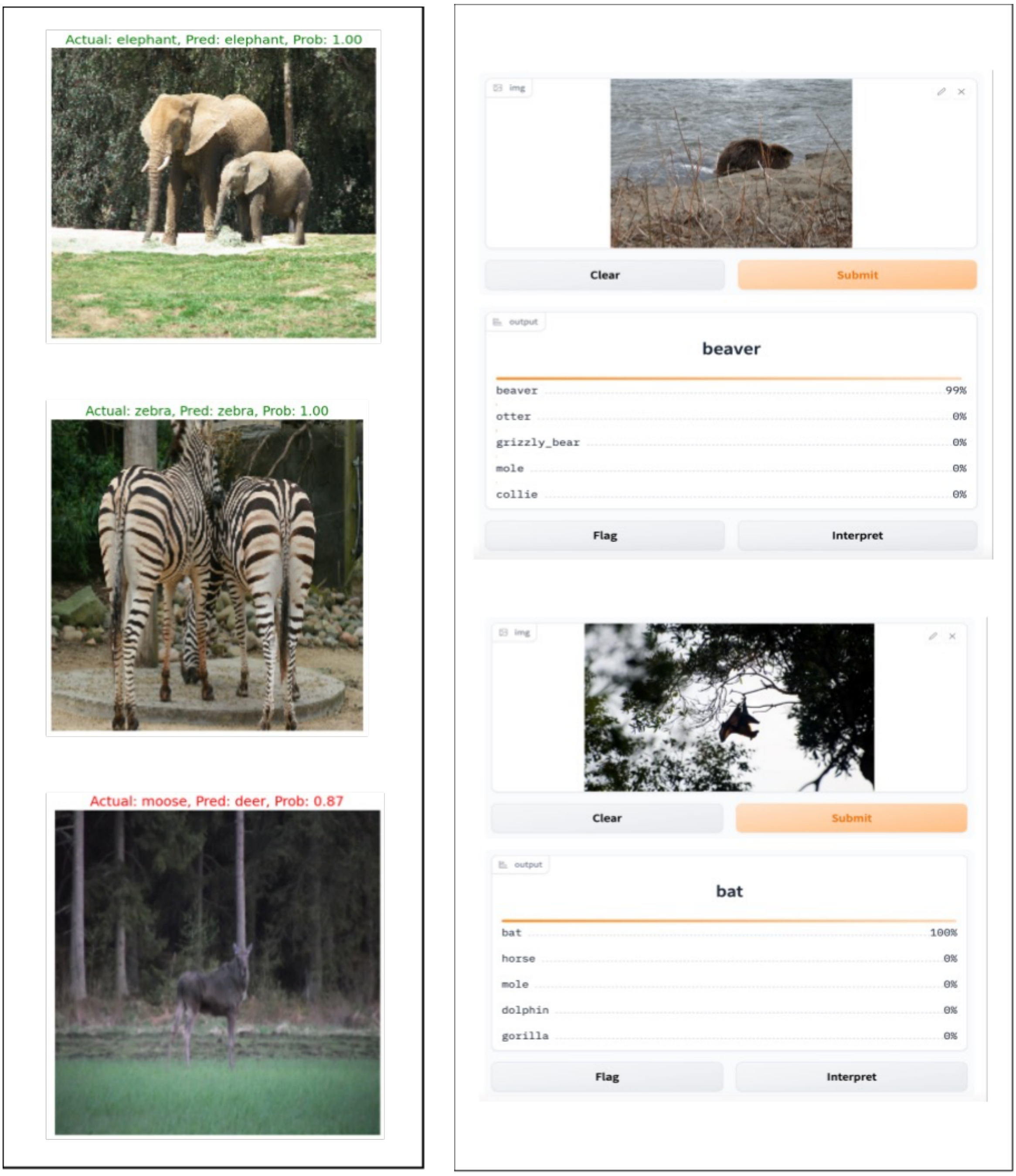


Fig. 7. Best model performance metrics (F1-score, Precision, Recall) for multi-species classification

3.2.3. Model prediction on test images

Our experiment provides evidence that our best model is highly accurate in classifying both known and unknown test images (as demonstrated in Figure 8a). Specifically, we found that the model was able to accurately classify the target images into appropriate classes, such as Elephant and Zebra, based on the actual image labels. The green highlights in the "actual," "pred" (predicted class), and "prob" (probability of predicted class with actual class) indicate the images correctly classified by our best model, while the red highlights indicate the misclassified images (Figure 8a). This is a critical result that shows that the model has learned to identify the unique features of each class, allowing it to generalize well to new, unseen images. In addition to our experiment, we tested our model's performance using the Gradio interface along with the Tkinter interface. The results obtained from the interfaces were consistent with our experiment, as the model accurately classified both known and unknown test images (as shown in Figure 8b). This is another important indicator of the model's reliability and robustness in image classification.



a) During Experimentation b) Gradio Interface

Fig. 8. Prediction of testing image datasets for model evaluation

The Tkinter interface evaluation involved the automatic classification of images in the testing datasets, including those that were previously unseen, into different class name folders based on the predictions generated by our models. For example, images of antelope and otter were correctly classified and sorted into their respective class name folders, demonstrating the accuracy and effectiveness of our best classification model. However, there was an instance where our model was unable to classify certain images. One such example was the Canadian lynx, which was not present in our model during the training phase. As a result, the Canadian lynx image was classified into the "Others" folder. It is important to note that this outcome does not necessarily indicate a failure on the part of our model, but rather highlights the limitations of its training data. Overall, the evaluation by the Tkinter interface provides valuable insights into the performance of our classification model and its ability to accurately classify images into the correct class name folders. It also underscores the importance of using comprehensive and diverse training data to ensure that the model is capable of accurately classifying a wide range of images, including those that may not be present in the initial training dataset (Figure 9).

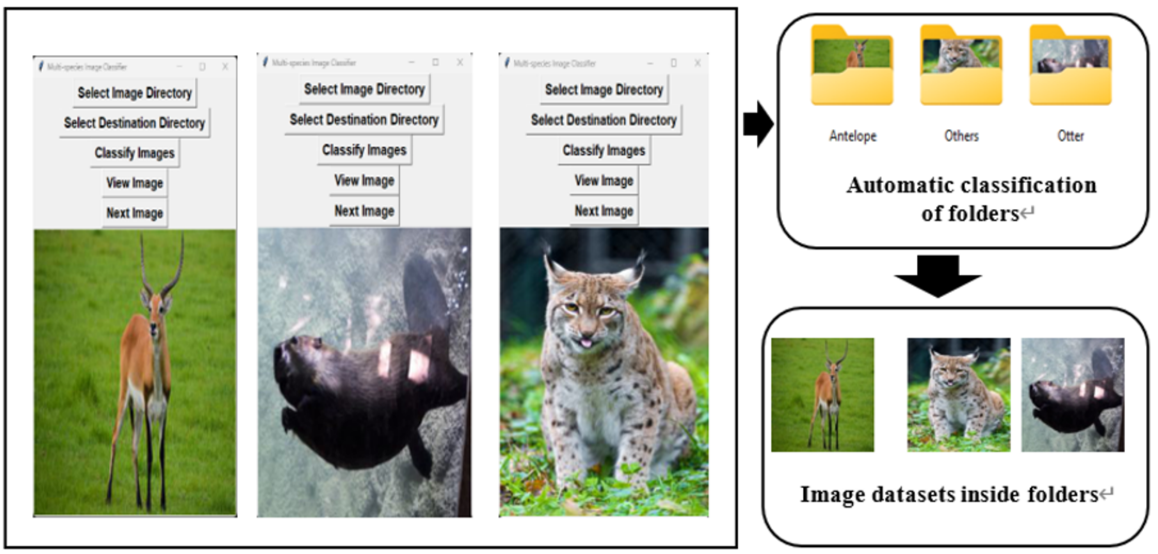


Fig. 9. Evaluation of previously unseen testing datasets using the Tkinter interface

4. DISCUSSION

The effectiveness of this study is based on its capability to automatically classify multiple species of wild animals, which has potential for assisting ecologists in automatically categorizing a large number of wildlife image datasets. Our results show that EfficientNetB0 reached high accuracies in multi-species classification compared with VGG16; however, these two deep-learning models are commonly used for image classification tasks. When they were trained with the same learning rate and activation function (Table 2), EfficientNetB0 achieved better accuracy than did VGG16, even though it had fewer parameters (EfficientNetB0 total parameters: 9,480,904; VGG16 total parameters: 14,722,624). The reason for this difference can be attributed to the architecture and parameter distribution of the models. EfficientNetB0 is designed with a more efficient architecture that balances depth, width, and resolution using a compound scaling method [16]. In contrast, VGG16 has a much deeper architecture with a larger number of non-trainable parameters (EfficientNetB0 non-trainable parameters: 14,722,624; VGG16 non-trainable parameters: 4,057,507), which may be more prone to overfitting [15]. On the other hand, due to the deeper architecture of VGG16 and its larger number of non-trainable parameters, it was more computationally expensive and slower to train, thus it required more time to train (Table 2).

In our study, we demonstrated that training an EfficientNetB0 model with a learning rate of 0.0001 and the ReLU activation function could achieve high accuracy and quality predictions on the image classification task, with both the accuracy and top-5 accuracy exceeding 90%. Our findings highlighted the efficiency and effectiveness of the compound scaling method used in the EfficientNetB0 design, which allows for competitive performance with a relatively small number of parameters. These results suggest that EfficientNetB0 is a strong candidate for image classification tasks where efficiency and accuracy are both important factors [16]. Despite this, our best image classification model achieved recall, precision, and F1-score values of over 0.90 for each class, which indicate high performance and accurate distinction between classes.

Our experimental results, as well as the performance of the Gradio and Tkinter interfaces, confirm the accurate classification of the model prediction on test image datasets. This indicates that the model successfully learned the distinguished features of each class and could generalize well to new instances [20]. Furthermore, the model's ability to generalize demonstrates that it did not overfit the training data, which is a common problem in machine learning. Overfitting occurs when the model becomes specific to the training data, which can result in poor performance when it is applied to new data. However, our best model managed to avoid overfitting, which is a significant achievement.

The present study investigates the effectiveness of using transfer learning techniques on pre-trained models for classifying images of wild animals with a limited training dataset. The results suggest that implementing data augmentation techniques can enhance the model's accuracy and mitigate the influence of the uneven distribution of image data among various species, thereby minimizing biases in the model's predictions. Additionally, the study underscores the significance of hyperparameter tuning, particularly the learning rate and activation functions, which play a crucial role in determining the magnitude of updates made to a model's performance during training, thereby achieving optimal performance. The findings of this study have critical implications for researchers and developers working in the domain of computer vision and image classification, particularly in the realm of wildlife conservation.

Our work presents a novel approach to image classification of wildlife species using transfer learning. This approach significantly reduces the time and effort required for manual classification. Previous machine learning methods classify images of the test or unseen wildlife species into different classes, but they do not organize the images into specific class name folders. Our approach, on the other hand, automatically classifies the test or unseen wildlife images and organizes them into different specific class name folders, which will greatly benefit wildlife researchers and conservationists. Creating different class name folders will ease the conservationist's work as they can easily identify and retrieve the images of interest. Our work stands out in its ability to accurately classify images and organize them into specific folders, which is a significant contribution to the field of wildlife image classification.

This study presents a promising approach for multi-species classification, yet it has several limitations. First, the approach was tested on a small subset of wild animal species, and it is unclear how effective it would be for a more extensive range of species. Second, although the proposed approach showed promising results for classifying wild animal images, it may not be directly applicable to other domains or tasks. Moreover, the approach required significant computational resources for training due to the use of pre-trained models, making it infeasible for organizations having limited computing capabilities. Finally, although the study provides valuable insights for researchers, further research and development are necessary before it can be applied in real-world applications, such as a web app for the automatic classification of multiple species.

5. CONCLUSION

In conclusion, we demonstrated the effectiveness of our deep-learning model in accurately identifying multiple wildlife species using a diverse set of standardized images. Our best model, which used EfficientNetB0 with ReLU activation functions and a learning rate of 0.0001, achieved a high accuracy of 92% and a top-5 accuracy of 99%, with F1-score, precision, and recall values exceeding 0.90 for all species categories. We found that our best model generalized well to previously unseen images, which we demonstrated using the Gradio and Tkinter interfaces. This highlights the model's robustness and reliability for image classification. By showcasing the capabilities of deep-learning models for the species classification task, our study offers valuable insights and resources for researchers and practitioners involved in wildlife conservation.

ACKNOWLEDGMENTS

We gratefully acknowledge the support provided by the Project base AI (PBL-AI) program and the Ministry of Education, Culture, Sports, Science, and Technology (Monbukagakusho: MEXT) scholarship awarded to Sandhya Sharma. We also thank NTT-Docomo for providing the International Student Scholarship to Sangam Babu Neupane. We would like to express our sincere appreciation to

Enago (www.enago.jp) for their invaluable assistance with the English language review of this manuscript. Finally, we declare that all authors have no conflicts of interest.

REFERENCES

1. O'Connell, AF, Nichols, JD, Karanth, KU 2011 'Camera traps in Animal Ecology: Methods and Analysis,' *Springe*. 10.1007/978-4-431-99495-4
2. Thomson, R, Potgieter, GC, Bahaa-el-din, L 2018, Closing the gap between camera trap software development and the user community, *African Journal of Ecology*, vol. 56, pp. 721-739. 10.1111/aje.12550
3. Young, S, Rode-Margono, J, Amin, R 2018, "Software to facilitate and streamline camera trap data management: A review," *Ecology and Evolution*, vol. 8, pp. 9947-9957. 10.1002/ece3.4464
4. Fennell, M, Beirne, C, Burton, AC 2022, "Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology," *Global Ecology and Conservation*, vol. 35
5. Yu, Q 2022, "Animal image classifier based on convolutional neural network," *SHS Web of Conferences*, vol. 144. 10.1051/shsconf/202214403017
6. Nguyen, H, Maclagan, SJ, Nguyen, TD, Nguyen, T, Flemons, P, Andrews, K, Ritchie, EG, Phung, D 2017, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring," vol. 2017 IEEE International Conference *on* Data Science *and* Advanced Analytics *(DSAA)*, Tokyo, Japan, pp. 40-49. 10.1109/DSAA.2017.31
7. Shiddieqy, HA, Hariadi, FI, Adiono, T 2017, "Implementation of deep-learning based image classification on single board computer," vol. 2017 International Symposium *on* Electronics *and* Smart Devices *(ISESD)*, Yogyakarta, Indonesia, pp. 133-137. 10.1109/ISESD.2017.8253319
8. Wei, W, Luo, G, Ran, J, Li, J 2020, "Zilong: A tool to identify empty images in camera-trap data," *Ecological Informatics*, vol. 55. 10.1016/j.ecoinf.2019.101021
9. Norouzzadeh, MS, Nguyen, A, Kosmala, M, Clune, J 2018, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *PNAS*, vol. 115, pp. E5716-E5725. 10.1073/pnas.1719367115
10. Lampert, CH, Nickisch, H, Harmeling, S 2009, "Learning to detect object classes by between class attributes transfer" *In* Proceeding *of the* IEEE Computer Society Conference *on* Computer Vision *and* Pattern Recognition *(CVPR 2009)*, Milami Beach, FL, USA, pp. 951-958. 10.1109/CVPR.2009.5206594
11. Lee, J, Lim, K, Cho, J 2022, "Improved monitoring of wildlife invasion through data augmentation by extract-append of a segmented entity," *Sensors*, vol. 22, p. 7383. 10.3390/s22197383
12. Georgopoulos, M, Oldfield, J, Nicolaou, MA, Panagakis, Y, Pantic, M 2021, "Mitigating demographic bias in facial datasets with style-based multi-attribute transfer," *International Journal of Computer Vision*, vol. 129, pp. 2288-2307
13. Jinyin, C, Yipeng, C, Yiming, C, Haibin, Z, Shouling, J, Jie, S, Yao, C 2021, "Fairness research on deep learning," *Journal of Computer Research and Development*, vol. 58, p. 264
14. Faizal, S, Sundaresan, S 2022, "Wild animal classifier using CNN," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 2, p. 2581-9429. 10.48175/IJARSCT-7097
15. Simonyan, K, Zisserman, A 2015, "Very deep convolutional networks for large-scale image recognition" *In* Proceedings of the 3rd international conference on learning representations (ICLR), San Diego, CA, USA, pp. 1-14

16. Tan, M, Le, QV 2019, "EfficientNet: Rethinking model scaling for Convolutional Neural Networks" *In Proceedings of the 36th international conference on* machine learning *(ICML)*, long Beach, California, vol. 97, pp. 6105-6114
17. Iqbal, S, Qureshi, AN, Ullah, A, Li, J, Mahmood, T 2022, "Improving the robustness and quality of biomedical CNN models through adaptive hyperparameter tuning," *Applied Sciences*, vol. 12, p. 11870. 10.3390/app122211870
18. Mujhid, A, Surono, S, Irsalinda, N, Thobirin, A 2022, "Comparison and combination of Leaky ReLU and ReLU activation function and three optimizers on deep CNN for COVID-19 detection" *In* Proceedings of the 8th international conference on fuzzy systems and data mining (FSDM), vol. 358, pp. 50-57
19. Kutugata, M, Baumgardt, J, Goolsby, JA, Racelis, AE 2021, "Automatic camera-trap classification using wildlife-specific deep learning in Nilgiri management," *Journal of Fish and Wildlife Management*, vol. 12, pp. 412-421. 10.3996/JFWM-20-076
20. Abid, A, Abdalla, A, Abid, A, Khan, D, Alfozan, A, Zou, J 2019, Gradio: Hassle-Free Sharing and Testing of ML models in the wild, "*arXiv*, vol. 1906.02569