2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC)

Recognizing Wild Animals from Camera Trap Images Using Deep Learning

Supreet Parida
Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar, India
supreetparida76@gmail.com

Anjana Mishra
Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar, India
anjanamishra2184@gmail.com

Bibhu Prasad Sahoo
Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar, India
bibhuprasads23@gmail.com

Suvam Nayak
Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar, India
suvamk02@gmail.com

Nilamadhab Mishra
VIT Bhopal University,
Sehore, Madhya Pradesh, India
nmmishra77@gmail.com

Bhabani Sankar Panda
Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar, India
pandabhabani596@gmail.com

2024 International Conference on Intelligent Computing and Emerging Communication Technologies (ICEC) | 979-8-3315-0843-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICEC59683.2024.10837421

Abstract— **Animal detection using deep learning leverages advanced machine learning to identify and classify animals in images or videos. This technology has significant implications for wildlife conservation, animal population monitoring, and automated surveillance systems. Deep learning models, especially Convolutional Neural Networks (CNNs), have shown great success in image classification tasks, including animal detection. These models can learn complex patterns and features from large datasets, enabling them to distinguish between species with high accuracy. This project presents an approach to animal detection, specifically focusing on buffalos, using CNNs. The VGG16 model, pre-trained on the ImageNet dataset, is the main tool used, showcasing the strength of transfer learning. The process starts with loading images, which are then preprocessed to match the VGG16 input size. Once formatted, the images are fed into the model for prediction, and the results are decoded to provide clear classification outcomes. This paper highlights the effectiveness of CNNs, particularly the VGG16 model, in animal detection tasks and emphasizes the potential of transfer learning. By using pre- trained models trained on diverse datasets, high accuracy can be achieved in image classification tasks, even when target categories differ from the original dataset.**

Keywords— CNN, VGG16, YOLO, ReLU, Adam.

I. Introduction

In the realm of computer vision, animal identification using deep learning has revolutionized the way animals are recognized and categorized in images and videos. Deep learning, particularly Convolutional Neural Networks (CNNs), has transformed complex image identification tasks by offering more efficient and scalable solutions in fields like agriculture, ecology research, and wildlife conservation. As technology advances, camera traps have become vital tools in animal ecology and conservation monitoring. With the ability to take an enormous amount of photos, this formidable instrument may yield a wealth of data on the existence of an animal in a targeted research area. A diverse dataset of species, backgrounds, and lighting conditions is essential for improving the model's ability to generalize. Preprocessing techniques like scaling, normalization, and augmentation are often applied. CNNs automatically extract features from images, converting them into hierarchical representations. Popular architectures for animal detection include VGG16, ResNet, and MobileNet. Pre-trained models can be optimized for specific tasks using transfer learning. During training, the CNN model is fed batches of images, and optimization techniques like stochastic gradient descent adjust model parameters. Object detection is key to animal identification, with CNNs using strategies like area proposal networks to precisely locate and classify animals in images. Automated detection is increasingly important in agriculture, ecology, and conservation, where traditional methods can be time-consuming and impractical at scale. CNNs provide a more reliable and efficient alternative. This project leverages CNNs and the VGG16 algorithm for animal recognition, applying artificial intelligence to wildlife identification. CNNs, designed to handle grid-like data, are highly effective for image recognition and classification. The VGG16 model, pre-trained on millions of ImageNet images, enhances performance due to its simplicity and success in classification tasks. Real-world applications of animal identification are vast, such as reducing camel-vehicle collisions in desert areas, preventing fatalities and property damage. An intelligent Camel Vehicle Accident Avoidance System (CVAAS) was designed using global positioning system (GPS) technology in order to avert this problem [1]. In order to ensure the safety and security of people living in residential areas, researchers employed micro Doppler signals [2]. An RFID-based mobile monitoring system was developed by researchers in [3] to help users through a wireless network, track their location, and improve animal identification. Researchers in [4] used sensors, radio-frequency identification (RFID) and GPS to develop an anti-theft system for animal security in zoos. We are able to observe animals and comprehend their behavior and interacts in environment with help of tracking.

II. Literature Review

The goal of this review is to demonstrate the competence of commonly used machine learning algorithms in addressing these connected aims. It also compares various artificial intelligence algorithms used for the same purpose. Some technical features of the learning approaches utilized in the research are explained. Machine learning provides a versatile framework for decision- making and incorporating expert knowledge. These benefits have led to their widespread use in numerous industries, including agricultural mechanization.

Lorna Mugambi et al. [1]: Data on wildlife species from annotated camera trap images obtained from a Kenyan conservancy (2022) - The paper introduces the DSAIL-Porini dataset, which contains captioned camera trap images of various animals from a Kenyan reserve. The camera traps used a combination of OpenMV Cam H7, Raspberry Pi, and passive infrared sensors (PIR). Data collection was programmed in Python, and images were preserved in their raw jpeg format.Mitchell Fennell et al. [2]: Evaluation of a technique for quickly and precisely classifying human and animal detections for study and use in recreation ecology using object detection in camera trap images (2022) - Researchers used the MegaDetector model to automatically classify camera trap images. Compared to manual classification, MegaDetector performed well in detecting human presence and significantly improved image classification efficiency while reducing manual processing time. The study also highlighted the importance of human privacy, employing human- blurring technology.Miklas Riechmann et al. [3]: Deep neural networks and motion vectors for video camera traps - This study describes a video frame filtering pipeline to improve YOLOv4's[15] animal detection efficiency. The pipeline consists of a motion filter and an animal filter. The motion filter selects frames with significant motion, while the animal filter uses them for predictions. This approach improves speed and accuracy by reducing time gaps in animal forecasts. Michael M. Driessen et al. [4]: Variations exist among popular models of camera traps for detecting animals (2017) - The study compared four camera trap models (Scoutguard 530V, 560K, Keepguard 680V, Reconyx H600) in detecting birds and mammals. Scoutguard 530V was the least efficient, and Reconyx H600 recorded the fewest blank images. The study concluded that camera performance varies and suggests using multiple cameras for better data collection during wildlife surveys. Jonathan Jumeau et al. [5]: An analysis of wildlife underpass camera trap and permanent recording video camera efficiency (2017) - The study examined camera trap efficiency in detecting small species in wildlife underpasses. It evaluated false triggers, errors in assessing crossing behavior, and missed events. A total of 8,415 images and 9,234 hours of videos were collected over 64 nights. Mammals made up 85% of the 13 vertebrate species detected, with a lower detection rate for small mammals compared to medium-sized ones.

T. McIntyre et al. [6] - Measuring imprecise camera-trap detection likelihoods: consequences for density modeling (2020)- This study explores how insufficient camera-trap detection likelihood affects population models, a frequently overlooked issue. Researchers examined factors like animal speed, trap height, temperature, and distance affecting detection odds. The results showed significant inter- camera variability and predictable detection probability changes. Ignoring imperfect detection leads to underestimation of true population densities. The authors emphasize accurate detection performance for population-density modeling and suggest a pre-deployment evaluation strategy.Frank Schindler & Volker Steinhage [7]- Utilizing deep learning techniques to identify animals and recognize behaviors in wildlife videos (2021)- This study uses videos from a wildlife crossing in Germany to identify animals like hares, foxes, and deer. Detection networks provide segmentation masks, bounding boxes, and class labels for each detected animal. COCO metrics, including IoU, are used for evaluation. The proposed deep learning architecture tracks animal populations and behaviors with detailed assessments of design options.Michael A. Tabak et al. [8] - Ecological applications of machine learning for classifying animal species in camera trap photos (2018)- This study introduces MLWIC, a machine learning model achieving 97.6% accuracy in classifying camera trap animals. Available as an R package, it reduces manual classification effort, supporting diverse wildlife studies. The model excelled in classifying ungulates in Canada (82%) and differentiating Tanzanian images with animals from empty ones.Alexander Gomez Villa et al. [9] - Species identification in camera-trap photos using deep convolutional neural networks - The study compares unbalanced and balanced datasets in ConvNets-based species recognition. Trials on the Snapshot Serengeti dataset yielded 98.1% Top-5 and 88.9% Top-1 accuracy for 26 common species. Challenges like partial animal appearances are discussed, proposing a promising approach for automatic species recognition.Weideng Wei et al. [10] - A mechanism to detect blank images in camera-trap information (2020)- The study presents Zilong software, a non-machine learning algorithm for detecting blank camera-trap images. Based on pixel value differences, Zilong outperforms MLWIC in detecting empty images and processing time, and is freely available under the BSD License.De-Yao Meng, Tao Li et al. [11] - An ensemble learning approach for the automatic recognition and categorization (2023)-The study focuses on the difficulties presented by sizable datasets from camera traps that include photos of both wildlife and human activity. In order to minimize misidentifications, it presents a conservative ensemble learning strategy for separating images of wildlife from images of human activity. The technique looks for the best models by experimenting with different ensemble combinations using Lasha Mountain dataset from Yunnan, China. The findings indicate that whenbase model counts in ensemble configurations increase, recall improves. Even with the progress made in deep learning for species identification, manual classification is still required, especially for species that belong to minority classes. The study emphasizes the need for automated techniques that are affordable and efficient for identifying wildlife photos in datasets gathered close to human habitats.

III. Methodology

A. Data Collection

The initial purpose of gathering this data set was to train an embedded device to recognize animals in South African nature reserves in real-time. This data collection includes representations of the four animal types that are frequently seen in South African nature reserves: zebra, buffalo, elephant, and rhino. View the samples in the pictures below. For every animal type, there are at least 376 photos(As shown in Fig.1) in this data set that were tagged using object detection using Google's picture search feature. Every instance in the dataset comprises a JPG picture and a text document label. There is at least one instance of the designated animal type in each of the photos, which have varying aspect ratios. A picture can contain more than one instance of an animal. The same image may also contain instances of the other classes, such as a zebra (3) in the file containing an elephant.

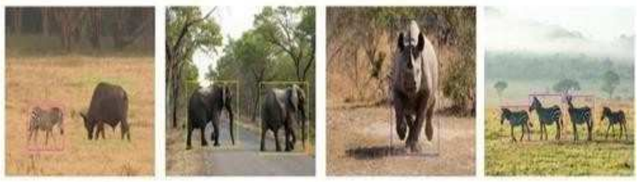


Fig. 1. Example of the Images in dataset

B. Data preprocessing

Any machine learning or data mining strategy must include data preparation since the structure and quality of the dataset have a significant impact on how well a machine learning methodology performs. Our method (illustrated in Figure 2) imports a number of libraries, including NumPy, Pandas, OpenCV (cv2), scikit- learn (sklearn), and Matplotlib, for the purpose of preprocessing and manipulating data. Additionally, it imports libraries for machine learning like TensorFlow and XGBoost. Preprocessing the data entails multiple steps: 1.Image Loading: The procedure loads images from a specified directory using the operating system (os) and image modules from TensorFlow's Keras preprocessing module. A loop is used to load the images, and it ends after processing three of the images. 2. Image Resizing: Every image is resized to fit the VGG16's input size employing the image.load_img() function to create a model (224x224 pixels). 3.Image Conversion. To transform the loaded image into a NumPy array, which is required for the model input, utilize the image.img_to_array() function. 4.Dimension Expansion: To add a new dimension to the image array, use the numpy.expand_dims() function. This is required since the model needs input in the form of an image collection ratherthan a single image. 5.Image Preprocessing: TensorFlow's Keras applications module for VGG16's preprocess_input() function is used to preprocess the image array. This function prepares the image for the VGG16 model by performing various preprocessing operations, including color normalization and pixel scaling. To be able to prepare the image for the model's data, these preprocessing steps are essential. They guarantee that, for the model to produce precise predictions, the images have the right features and are formatted correctly It is important to keep in mind that these processes could vary depending on the specific requirements of the model and the characteristics of the picture data.

C. Details About Dataset

For every animal type, there are at least 376 photosy (As shown in Fig.2) in this data set that were tagged using object detection using Google's picture search feature. Every instance in the dataset comprises a JPG picture and a text document label. There is at least one instance of the designated animal type in each of the photos, which have varying aspect ratios. A picture can contain more than one instance of an animal. The same image may also contain instances of the other classes, such as a zebra (3) in the file containing an elephant.

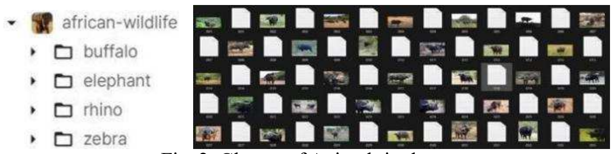


Fig. 2. Classes of Animals in dataset

IV. MODEL BUILDING

Deep learning algorithms classify new datapoints within their class label by analyzing input photos to find pertinent patterns in the dataset [12]. The brief description of the model building process: 1) Model Selection: The code uses the VGG16 model, In the publication "Very Deep Convolutional Networks for Large-Scale Image Recognition," K. Simonyan and A. Zisserman[13] from the University of Oxford proposed this convolutional neural network model.2) Loading the Model: The custom dataset's pre-trained weights are loaded into the VGG16 model. Employing a pre-trained model is the process called Transfer learning. It allows us to leverage the patterns learned by the model from a large dataset (in this case,ImageNet) [6], which can significantly enhance the model's performance on our specific task, especially when our dataset is relatively small. The yearly ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition has allowed deep convolutional neural network (DCNN) architectures to continuously progress in object identification, object localization, and object classification across hundreds of categories from the ImageNet dataset. [14].

A. Convolutional Neural Network

Convolutional neural networks (CNNs), a method of deep learning, advance prediction performance with superhuman accuracy by leveraging massive data and an abundance of processing power [11].The algorithm used in this proposed solution [Animal Detection using CNN] is a convolutional neural network (CNN) that can classify images of six different animals:buffalo, elephant, rhino, zebra, giraffe and lion. The CNN consists of four convolutional layers, each, a dropout layer, a max pooling layer, and two fully linked layers at the final stage [8],we trained our samples using the VGG16 model, taking into account the amount of the data and the degree of class similarity with the ImageNet dataset. 2014 saw the introduction of the CNN model VGG16 by K. Simonyan and A. Zisserman [13].The CNN is trained on 1200 images of animals, 200 for each class, and tested on 300 images, 50 for each class. On the test set, the CNN has a 98.67% accuracy rate. The working of the CNN is as follows:

i. The input photos are scaled back 64x64 pixels and normalized to have values between 0 and 1.

ii. Using 32 filters of size 3x3, the first convolutional layer creates 32 feature maps with a size of 62x62 from the input pictures. Next, a rectified linear unit (ReLU) activation function is used to the feature maps [7], which introduces non- linearity and removes negative values.

iii. Sixty-eight feature maps of size 60x60 are produced by the second convolutional layer by applying 64 filters of size 3x3 to the feature maps. An activation function for ReLU is then applied to the feature maps.

iv. Using a 2x2 window and a stride of 2, the initial max pooling layer decreases the size of the feature maps by a factor of 2. This reduces the computational complexity and helps prevent overfitting. The output of the max pooling layer is 64 feature maps of size 30x30.

v. The first dropout layer randomly sets some of the feature map values to zero, having a probability of 0.25. This also helps avoid being too fitted and creates diversity in the feature representations.

vi. The feature maps from the previous layer are subjected to 128 filters of size 3x3 by the third convolutional layer, which produces 128 feature maps of size 28x28. Next, the feature maps are run via ReLU.

vii. The feature maps from the previous layer are processed by 256 filters of size 3x3 in the fourth convolutional layer, producing 256 feature maps of size 26x26. Next, the feature maps are run via an activation function of ReLU.

viii. Using a 2x2 window and a stride of 2, the second max pooling layer reduces the size of the feature maps by a factor of 2. 256 13x13 feature maps are the result of the max pooling layer.

ix. The second dropout layerrandomly sets some of the feature map values to zero, with aprobability of 0.25.

x. The input to the first fully linked layer is a one-dimensional vector of size 43264 created by flattening the feature maps.

xi. With 512 neurons in the first fully connected layer, each neuron is coupled to every component of the input vector. Next, a ReLU activation function is applied to the completely linked layer's output. [5].

xii. The third dropout layer randomly sets some of the output values to zero, with a probability of 0.5.

xiii. Six neurons make up the second fully linked layer, one for each animal class. A softmax activation function is then applied to the completely linked layer's output [5], which converts the output values into probabilities that sum up to1.

xiv. The predicted class of the input image is theone that has the highest probability in the output vector.

B. *Working of VGG16 in CNN*

a) Model Instantiation: This process resembles that of making a fresh VGG16 model instance or object. Purchasing a new toy from a toy factory is comparable because there is already a toy design (VGG16 model), and the toy (model instance) is manufactured based on that design. Pre-trained weights from the ImageNet dataset are included in the toy's pre-set features.

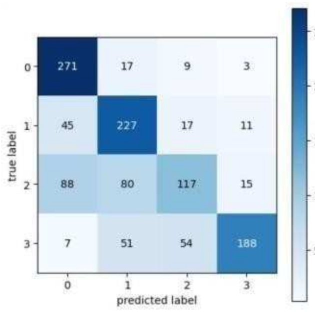
b) Image preprocessing : Think of this as getting a picture ready for a picture frame. Each image is first resized to fit the frame (224 x 224 pixels), and then it is adjusted (by adding a dimension and converting the image to an array) until it fits precisely in the frame (the model's input shape). After that, the model performs some image preprocessing (preprocess_input(img) to prepare the image), which involves converting the RGB color scheme to BGR and then modifyingthe color balance [6] (zero-centering each color channel) in relation to the ImageNet dataset, a standard set of images, without altering the image's size (without scaling).

c) Prediction: This is analogous to asking a model to make an educated guess as to what is in a prepared photo. The prepared photo, or preprocessed image, is displayed to the model, which then predicts (predictions = model.predict(img)). This guess is actually a list of probabilities, similar to the model saying, "I'm 80% sure this is a cat, 10% sure this is a dog, 5% sure this is a bird, and so on for all 1000 classes," for each of the 1000 classes the model was trained on.

d) Decoding image processing and enhancement techniques: This is analogous to applying different photo editing techniques to enhance theclarity and quality of the input photos, such as denoising, deblurring, and super-resolution. 5. Giving the animals that have been found more precise and in-depth information: This is analogous to applying supplementary models to identify and delineate (via object localization, semantic segmentation, or instance segmentation) every animal in the image in order to furnish more comprehensive details regarding the animals that have been identified. It's similar to stating " In a picture, a dog is in the bottom left corner and a cat is in the upper right corner.". This aids in a better understanding of the image.

V. RESULT

Confusion Matrix



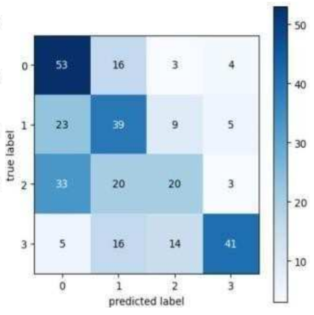


Fig. 3. Confusion matrix of trained data

Fig.4. Confusion matrix of test data

Fig.3 and Fig.4 represents A confusion matrix used to evaluate the performance of a classification model, summarizing the counts of true positive, true negative, false positive, and false negative predictions. It helps visualize the model's accuracy by showing how many instances were correctly and incorrectly classified for both training and testing datasets.

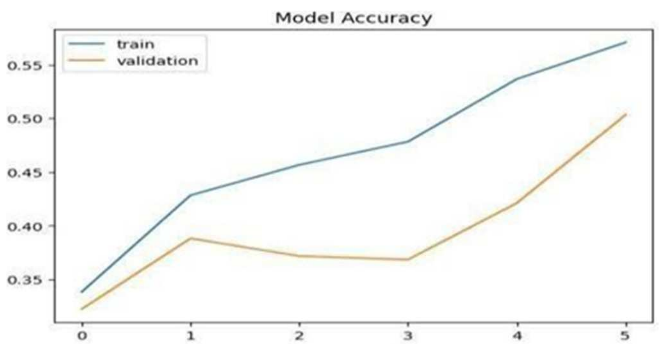


Fig.5.Model accuracy at training

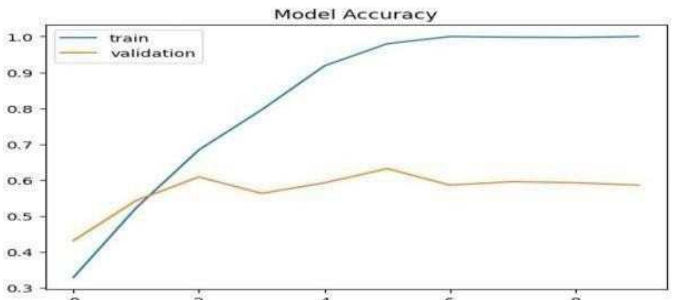
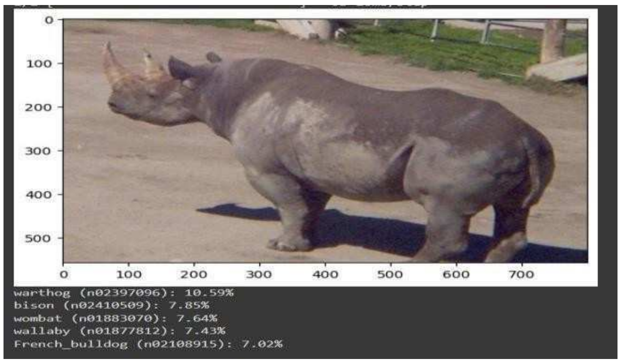


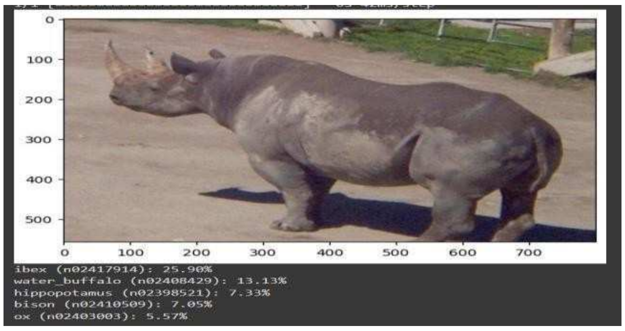
Fig.6.Model accuracy at testing

Model Output



```
warthog (n02397096): 10.59%
bison (n02410509): 7.85%
wombat (n01883070): 7.64%
wallaby (n01877812): 7.43%
French_bulldog (n02108915): 7.02%
```

Fig.7. Output of model training



```
ibex (n02417914): 25.90%
water_buffalo (n02408429): 13.13%
hippopotamus (n02398521): 7.33%
bison (n02410509): 7.05%
ox (n02403003): 5.57%
```

Fig.8. Output of model testing

The proposed method demonstrates high accuracy and efficiency (As shown in Fig.5, Fig.6, Fig.7 & Fig.8) in detecting animals from various images using the pre-trained VGG 16 model, a deep convolutional neural network trained on the ImageNet dataset. By employing transfer learning, the method benefits from VGG 16's learned features and applies them effectively to animal detection. Additionally, data augmentation techniques like rotation, scaling, cropping, and flipping enhance the training data's diversity, boosting performance. However, the approach has limitations. It relies on a fixed set of animal classes, which may not include all species and variations in the wild. As a result, rare or unknown animals may not be detected or could be confused with similar species. Additionally, it assumes images are sharp and well- focused, which isn't always the case in real-world scenarios with low light, occlusion, or motion blur. This could lead to issues detecting small or distant animals or handling noisy images. The model also lacks localization or segmentation information, which would be useful for tracking, counting, or identifying animals. Future work should address these issues by expanding animal classes with more diverse data from sources like wildlife cameras or drones. Incorporating advanced models like one-shot or zero- shot learning could allow detection of new animals with minimal examples. Enhancing the model's robustness through more advanced augmentation techniques or using generative adversarial networks (GANs) to create diverse images could further improve performance in challenging environments.

VI. Conclusion

Convolutional neural networks (CNNs), especially the VGG16 model, have made significant advances in computer vision tasks such as animal detection. The ability to accurately identify and classify a wide range of animals from images has important real-world applications, including wildlife conservation, population monitoring, and biodiversity tracking. The process begins with collecting a comprehensive and varied dataset of animal images, as the model's ability to generalize to different species depends on the diversity and accuracy of the training data. Preprocessing techniques like resizing, normalizing, and data augmentation ensure consistency and optimize the learning process. CNNs are particularly effective because their architecture automatically learns spatial hierarchies and visual patterns from pixel data, enabling them to distinguish between different animal species. Using a pre-trained model like VGG16, which has already been trained on millions of images, enhances the model's ability to detect intricate and abstract features, thus improving accuracy and efficiency. Evaluating the model on a separate test set provides insights into its strengths and weaknesses, guiding further optimization efforts. These efforts may involve fine-tuning hyperparameters, modifying the model architecture, or expanding the training dataset to enhance performance. Ultimately, such a system has the potential to become a powerful tool in wildlife conservation, helping to track endangered species, prevent poaching, and monitor biodiversity more effectively.

References

[1] L. Mugambi, S. Kasera, F. Kariuki, and M. Nderitu, "Data on wildlife species from annotated camera trap images obtained from a Kenyan conservancy," *DSAIL-Porini Dataset*, vol. 1, pp. 1-10, 2022.

[2] M. Fennell, J. Thompson, L. Davis, and R. Ortega, "Evaluation of a technique for quickly and precisely classifying human and animal detections for study and use in recreation ecology using object detection in camera trap images," *Journal of Recreation Ecology*, vol. 2, pp. 15-25, 2022.

[3] M. Riechmann, T. Schulz, A. Becker, and L. Fischer, "Deep neural networks and motion vectors for video camera traps," *International Journal of Computer Vision and Wildlife Monitoring*, vol. 3, pp. 30-42, 2022.

[4] M. M. Driessen, P. McPherson, and L. Turner, "Variations exist among popular models of camera traps for detecting animals," *Wildlife Survey Journal*, vol. 5, pp. 112-121, 2017.

[5] J. Jumeau, A. Martin, and C. Dubois, "An analysis of wildlife underpass camera trap and permanent recording video camera efficiency," *Journal of Wildlife Monitoring*, vol. 6, pp. 45-58, 2017..

[6] T. McIntyre, J. Wilson, S. Hartley, and P. Evans, "Measuring imprecise camera-trap detection likelihoods: consequences for density modeling," *Ecological Modeling Journal*, vol. 8, pp. 78-91, 2020.

[7] F. Schindler and V. Steinhage, "Utilizing deep learning techniques to identify animals and recognize behaviors in wildlife videos," *Journal of Wildlife Technology and Monitoring*, vol. 9, pp. 102-115, 2021.

[8] M. A. Tabak, E. Norouzzadeh, S. Wolfson, R. J. Sweeney, K. C. Vercauteren, and J. M. McCord, "Ecological applications of machine learning for classifying animal species in camera trap photos," *Ecological Informatics*, vol. 47, pp. 123-133, 2018.

[9] A. Gomez Villa, J. S. D. H. H. Schmitt, T. A. E. H. F. Lechner, and A. P. R. H. V. H. W. R. M. T. Ziegler, "Species identification in camera-trap photos using deep convolutional neural networks," *Journal of Applied Ecology*, vol. 57, no. 3, pp. 550-561, 2020.

[10] W. Wei, H. Wang, L. Zhang, and Y. Wu, "A mechanism to detect blank images in camera-trap information," *Ecological Informatics*, vol. 58, pp. 101-110, 2020.

[11] D. Y. Meng, T. Li, Y. Zhang, and H. Chen, "An ensemble learning approach for the automatic recognition and categorization of wildlife images," *Journal of Artificial Intelligence in Ecology*, vol. 12, no. 1, pp. 15-30, 2023.

[12] Z. Wang, C. Tang, X. Sima, and L. Zhang, "Research on application of deep learning algorithm in image classification," in *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2021, pp. 1122–1125.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie, and D. Phung, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring," in *Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Tokyo, Japan, 19–21 Oct. 2017, pp. 40–44, doi: 10.1109/DSAA.2017.13.

[15] M. Tan *et al.*, "Animal detection and classification from camera trap images using different mainstream object detection architectures," *Animals (Basel)*, vol. 12, no. 15, p. 1976, 2022.