Classifying 500 Bird Species using Pre-Trained CNN Models

Puneeth Nagarajaiah
CSML1020
York University
puneeth@my.yorku.ca

David Geller
CSML1020
York University
dgeller@my.yorku.ca

***Abstract*—*This report presents an image classification system for identifying different bird species from images. The system is based on a convolutional neural network (CNN) architecture, specifically MobileNet-V2 and EfficientNet-B0 pre-trained models. The dataset consists of over 500 bird species and contains 80,0085 images.*

Keywords—CNN, Tensorflow, Keras, EfficientNet, MobileNet

I. Introduction

Bird species classification is a challenging task due to the high variability in bird appearance, pose, and lighting conditions. However, recent advances in deep learning techniques have made it possible to achieve high accuracy in bird species classification using convolutional neural networks (CNNs). In this report, we present an approach for classifying images of 500 bird species using Mobilenetv2 and EfficientNetB0, two CNN architectures for image classification.

We utilized transfer learning to adapt the models to the bird species dataset and employed data augmentation techniques to increase the size of the dataset. Additionally, we utilized categorical cross-entropy as the loss function during training, and we used freezing and fine-tuning of layers to optimize the models' performance. The aim of this study is to evaluate the effectiveness of these models in classifying bird species images and to provide insights into their strengths and weaknesses.

The contribution of this report is in demonstrating the potential of Mobilenet-V2 and EfficientNet-B0 in improving the accuracy of bird species classification on a large dataset of 500 bird species. The results of this study could provide valuable insights for future research in the field of bird species classification and serve as a foundation for developing real-world applications, such as wildlife monitoring and bird species conservation efforts.

II. Related Works

In a study by Harjoseputro et al (2020), MobileNetV2 was used for bird species classification on a dataset of 200 bird species. They found that the standard CNN model provide a training accuracy of 98.38% and a testing accuracy of 72.38%. While on MobileNet, the result was a training accuracy of 96.27% and a testing accuracy of 70.59%. The authors concluded that the standard CNN had better accuracy than MobileNet, at the cost of a much larger model size[1].

Similarly, in a study by Sitepu et al, MobileNet, EfficientNetB0 and EfficientNetB3 models were trained and compared on a dataset of 200 classes of birds. The models achieved a validation accuracy of 76.38%, 81.475% and 84.021% respectively on the validation data and 75.30%, 80.73% and 81.30% on the test data [2].

III. Exploratory Data Analysis

The bird dataset that will be used to train the models consists of 80,085 training images, 2,500 test images and 2,500 validation images, covering 500 different bird species. The dataset is of high quality, with each image containing only one bird, which takes up at least 50% of the pixels in the image. The images are in JPG format and are of size 224 X 224 X 3. The dataset is structured into three subsets - training, test, and validation - each containing 475 subdirectories, one for each bird species. The images were hand-selected and pre-processed, with duplicates removed and cropping performed to ensure that the bird occupies at least 50% of the pixel space. However, the dataset is imbalanced with regards to male and female species images, and the test and validation images are predominantly of the male of the species. Despite this shortcoming, a moderately complex model should achieve training and test accuracies in the mid-90% range, making this dataset highly suitable for bird species classification tasks.

Figure 1a shows spread of Train Dataset

.Figure 1b shows spread of Train Dataset grouped by species

The species Red Tailed Thrush, Patagonian Sierra Finch and Snowy Plover have 130 images each, which is the minimum number of images per species, and the species House Finch has the highest number of images - 248, which is the maximum number of images for any species. Most of the species have 130-180 images. Nearly 20 species have more than 200 images. This points to a slightly imbalanced dataset with a 1:2 ratio between the minimum and maximum images per species.

IV. Model Architecture

A. MobileNet

MobileNet is a compact and computationally efficient deep neural network architecture designed for mobile and embedded vision applications. The key innovation of MobileNet is the use of depth-wise separable convolutions, which factorize a standard convolution into a depth wise convolution and a pointwise convolution, resulting in a significant reduction in the number of parameters and computations required while maintaining high accuracy on image classification tasks [3].

B. EfficientNet

EfficientNet is a deep neural network architecture that was introduced by Tan et al. in 2019. It is based on a scaling method that uses compound scaling to balance model depth, width, and resolution to optimize the overall model efficiency. The architecture includes a set of building blocks, called MBConv blocks, which consist of a combination of depth-wise separable convolutions, squeeze-and-excitation blocks, and inverted residual connections [4].

EfficientNet models are trained using a combination of image augmentation techniques and stochastic depth regularization and have been shown to achieve state of the art performance on a wide range of computer vision tasks while using significantly fewer parameters and computations than previous models of comparable accuracy.

V. Preparation For Training

A. Normalization

Normalization for images is a technique used in image processing to preprocess the pixel values of an image, such that they have a standardized scale and distribution, typically with a mean of zero and a standard deviation of one, in order to make it easier for machine learning models to learn and generalize patterns from the images.

The MobileNetV2 model expects an input of [-1, 1], so the RGB values (pixels) had to be rescaled using a built in rescaling function to prepare the images for training. On the other hand, the EfficientNetB0 model expects an input of [0, 255] and did not have to be rescaled due to the images already being in this range.

B. One Hot Encoding

One hot encoding for image classification is a technique used to represent the categorical labels of images as binary vectors, where each vector has a length equal to the number of classes, and only the index corresponding to the true class label is set to 1, while all other indices are set to 0, allowing for the categorical labels to be easily fed into a machine learning model during training and evaluation. This will be necessary for the categorical entropy loss function.

For the bird train, validation and test datasets, labels were specified as categorical. This was done by setting labels='categorical' parameter in the image_dataset_from_directory function in TensorFlow. This specifies the format of the labels returned by the function as a categorical array, which is a binary matrix representation of the class labels, rather than an array of integers. In this format, each label is represented by a vector of zeros except for a 1 in the index corresponding to the true class label, allowing for easy training of multi-class classification models.

C. Shuffling

The shuffle parameter in the image_dataset_from_directory function in TensorFlow is a Boolean value that determines whether to shuffle the order of the images within each class directory or not, before batching them into the dataset, which can help to prevent the model from learning any spurious correlations or biases that may exist in the ordering of the data. If set to True, the images are randomly shuffled within each class directory at the beginning of each epoch. In this case, only the train and validation datasets were set to true.

D. Data Augmentation

Data augmentation is a technique used in machine learning to artificially increase the size of a training dataset by applying various transformations to the original data. These transformations can include image rotation, flipping, zooming, or cropping, and are designed to help improve the performance and generalization of machine learning models.

For both models, the following data augmentations were performed using the keras.layers library:

- **Random rotation:** randomly rotates the input image by a factor of 0.2 radians.
- **Random translation:** randomly translates the input image along both the height and width dimensions by up to 10% of the image size.
- **Random zoom**: randomly zooms into or out of the input image along both the height and width dimensions by up to 10% of the image size.
- **Random flip:** randomly flips the input image horizontally.
- **Random contrast:** randomly adjusts the contrast of the input image by a factor of 0.1.

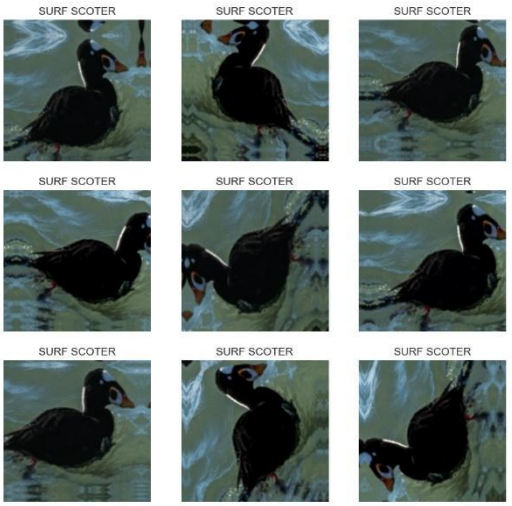

Figure 2 Showing the first 9 Augmented Images after the Data Augmentation was completed.

E. Other Considerations

Batch size was set to 32 and image size was set to 224 by 224 pixels.

VI. Training the models

A. Learning Rate

Learning rate is a hyperparameter that determines the step size at which a model updates its parameters during training. It controls how quickly or slowly the model learns from the data.

A high learning rate means that the model is updating its parameters rapidly, which can lead to overshooting the optimal values and cause instability. On the other hand, a low learning rate means that the model is taking small steps towards the optimal values, which can result in slow convergence or getting stuck in local minima.

After careful consideration and multiple test runs, the learning rate was selected to be 0.0001 with the Adam optimizer being chosen.

B. Categorical Cross Entropy

Categorical cross entropy is a loss function that measures the difference between the predicted probability distribution and the true probability distribution of the classes. It is often used in multiclass classification problems, where the output can belong to one of several classes [5].

Softmax activation is a mathematical function that transforms a vector of numbers into a probability distribution that sums up to 1. In the context of neural networks, softmax is often used as the activation function for the output layer. The softmax function takes a vector of inputs and applies an exponential function to each element, then normalizes the resulting vector so that the sum of its elements is equal to 1 [5].

$$L_{\text{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \text{ for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

Figure 3 Formula for Cross Entropy Loss Function [5]

In a neural network, the softmax function is typically applied to the output layer, producing a probability distribution over the possible classes. The categorical cross entropy loss function can then be used to measure the difference between the predicted probability distribution and the true probability distribution of the classes, providing feedback for the network to adjust its weights and biases during training.

MobileNetV2 and EfficientNetB0 are both using categorical cross entropy for its loss functions. This is available in both models libraries in TensorFlow.

C. Freezing Layers

In Brock et Al, a technique called FreezeOut is presented in order to accelerate the training of models [6]. The term "freezing" layers means keeping the weights of certain pre-trained layers fixed and not updating them during the training of the new model.

This is typically done when using a pre-trained model as a feature extractor, where the initial layers of the pre-trained model can be used to extract useful features from the input images. The later layers of the pre-trained model, which are more specialized and specific to the original task the model was trained on, can then be replaced with new layers that are specific to the new task at hand.

Freezing the weights of previous layers preserves the features they learned, enabling the training of new layers on extracted features without interfering with the earlier layers, often leading to improved performance and faster training.

D. Transfer Learning

Transfer learning is a technique in machine learning where a pre-trained model is used as a starting point for a new task, rather than training a model from scratch. The idea is to use the knowledge that a model has learned from a large amount of data on a related task and apply it to a new task.

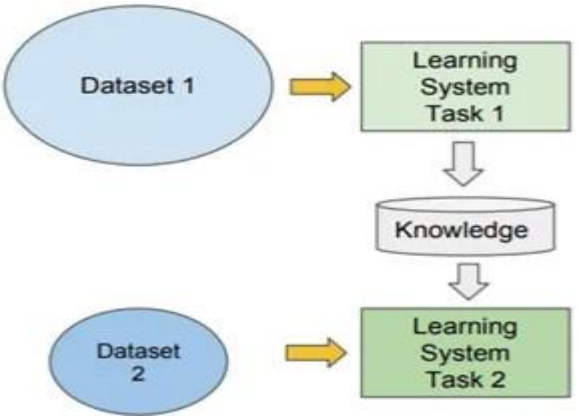

Figure 4 depicts the process behind Transfer Learning [7]

In transfer learning, the pre-trained model acts as a feature extractor for the new task, and the last few layers of the pre-trained model are replaced or fine-tuned to suit the new task. This approach can save significant time and resources in training a new model, especially when the new task has limited data available.

Both models use this technique by pre-training for 10 Epochs of the model and then employ freezing layers on the first 100 layers on the next 10 Epochs to fine tune the model. This yields better results for both models and accelerates training time.

	MobileNetV2	EfficientNetB0
# of layers before freezing	154	237
# of layers after freezing	54	137

Figure 5 Shows number of layers before freezing and after freezing

VII Results

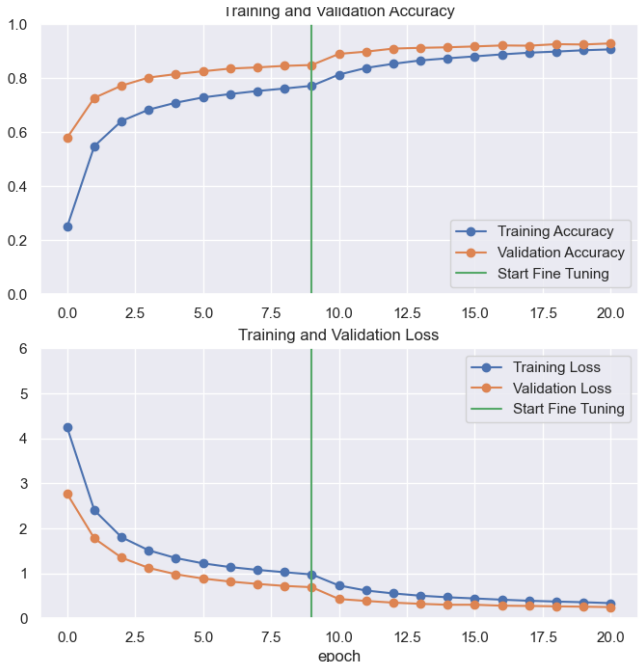

Figure 6a shows Training and Validation Accuracy and Loss Curves for fine-tuned MobileNet-V2 model

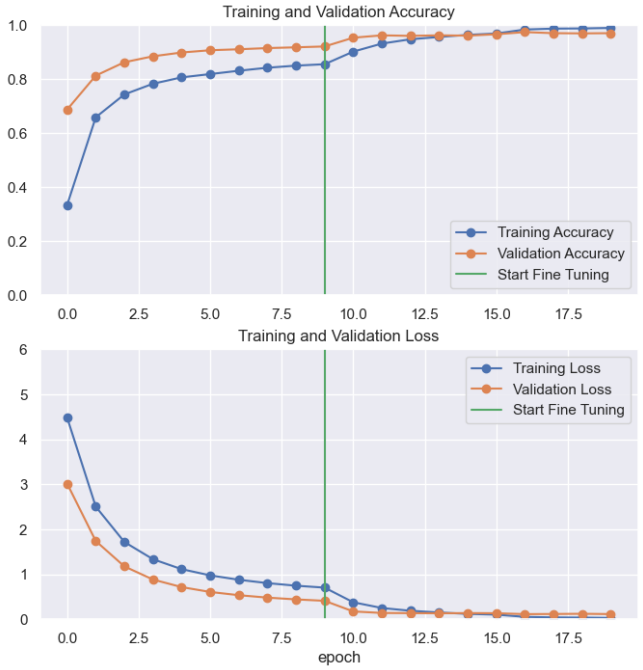

Figure 6b shows Training and Validation Accuracy and Loss Curves for fine-tuned EfficientNet-B0 model

	Accuracy	Precion	Recall	F1-Score
Mobile Net-V2	0.9472	0.96	0.95	0.94
Efficient Net-B0	0.9856	0.99	0.99	0.99

Figure 6c shows Accuracy, Precision, Recall and F1-Score for fine-tuned models

VIII Conclusion

In conclusion, the bird image classification project using MobileNetV2 and EfficientNet-B0 models showed that transfer learning can be a powerful technique for achieving high accuracy. The models were trained on a dataset of bird images and achieved high accuracy scores on the test set. MobileNetV2, being a lighter model, was able to achieve a reasonable accuracy of around 95% with fewer parameters and faster inference times. On the other hand, EfficientNet-B0, being a more complex model, achieved a very high accuracy of around 99% but with more parameters and slower inference times. Also, there seems to be some overfitting with respect to the EfficientNet model after the 6th epoch during the fine-tuning phase, which could be due to the slight imbalance in the dataset. This can be overcome by modifying the cross entropy loss function using class weights [8].

IX References

[1] Y. Harjoseputro, I. P. Yuda, and K. P. Danukusumo, “MobileNets: Efficient Convolutional Neural Network for Identification of Protected Birds,” International Journal on Advanced Science Engineering Information Technology. [Online]. Available: https://pdfs.semanticscholar.org/b9ae/6c3cbc3d6666df8c00448bed7aa4673b1c60.pdf.

[2] Sitepu, A. C., Liu, C.-M., Sigiro, M., Panjaitan, J., & Copa, V. (2022). A convolutional neural network bird’s classification using north American bird images. International Journal of Health Sciences, 6(S2), 15067–15080. https://doi.org/10.53730/ijhs.v6nS2.8988

[3] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861 [cs], Apr. 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[4] M. Tan, Q. Le, and others, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2019, pp. 6105-6114. [Online]. Available: https://arxiv.org/abs/1905.11946

[5]K. E. Koech. “Cross-Entropy Loss Function,” Towards Data Science. (2019). [Online]. Available: https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e#:~:text=Categorical%20cross%2Dentropy%20is%20used,%5D%20for%203%2Dclass%20problem.

[6] A. Brock, T. Lim and others. “Freezeout: Accelerate Training by Progressively Freezing Layers” (2017). [Online]. Available: https://arxiv.org/pdf/1706.04983.pdf

[7] Article by Dipanjan (DJ) Sarkar https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

[8] Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses by Trong Huy Phan and Kazuma Yamamoto https://arxiv.org/abs/2006.01413