**Bird Species Classification Using Convolutional Neural Networks and Vision Transformers**

By: Mrudav Patel & Kundan Kumar

**Introduction:**

In this project we develop several deep learning models to do bird species classification on a dataset with 525 classes and 84,635 images. CNN and Vision Transformer models are trained from scratch as well as using MobileNet, ResNet50, and ResNet101 pre-trained models for transfer learning. By comparing the training time and accuracies for the training, validation, and testing sets resulting from each model, we are able to understand our dataset better and what architectures are well suited to this species classification task.

**Background:**

Image classification using machine learning, also known as computer vision, has been an active field of study since the 1960’s. Its uses are wide ranging from medical diagnoses to autonomous driving and the architecture can take many different shapes such as convolutional neural networks, autoencoders, transformers, and more (Voulodimos et al. 2019). Specifically, classifying animals into different species using images or video is an area of research that has many interesting facets to it. Various objectives of animal classification include predator identification (Alharbi et al. 2018) and large-scale biodiversity studies (Christin et al. 2019).

Within the focus of this paper, there has been significant progress made in terms of bird classification by the Cornell Lab of Ornithology and their app, Merlin (Merlin Bird Id). Merlin is an app that lets the user attempt to identify a bird one of three ways, by picture, by description, or by sound; we will only be focusing on the picture identification method. While this method is useful for quick identification of birds, it does have drawbacks. Mainly, the picture needs to be both high quality and high resolution to obtain accurate results, additionally, the app relies on external data such as the date and location of the picture to help enhance the prediction. The extra data is useful for generating the most accurate result but could reduce effectiveness when the data is not available leading to a less robust model. This work will attempt to complete the classification task without additional data not present within the image itself.

**Method:**

The goal of our project is to classify the images of the birds by their species and sex. Classification of animals is an important scientific pursuit and is necessary to obtain accurate population counts and distribution. By having these statistics, researchers are able to better monitor individual species and ecosystems as a whole to determine their health and decide if any action is necessary to help these communities.

Being able to classify birds by species is a skill that takes a long time to master and the accuracy is dependent on what species an individual is exposed to over their bird watching career. By training a computer to be the classifier instead of a human, more accurate results can be achieved as well as a greater quantity of classifications. However, training a model to correctly identify species is difficult and relies on high quality data, complex model architecture, and ample training time.

To attempt to solve our problem, we developed a variety of different model architectures in order to determine what approach would yield the best results. Our models were divided into two categories, convolutional neural networks (CNN) and vision transformers. Furthermore, each of those categories are further subdivided into models we developed from scratch and pretrained models that we utilized transfer learning for.

All models utilized an input shape of 64x64 pixels which is a downsampling from the original image resolution of 224x224. This downsampling was performed to reduce the computational power necessary to train the model. Due to the lack of a GPU, multiple decisions were made that improved training efficiency but could negatively impact the model accuracy. The optimizer chosen was Adam with a learning rate of 0.0001, categorical accuracy was used to evaluate the model, and categorical cross entropy was used as the loss function.

The CNN models built from scratch consisted of two convolutional layers with 32 and 64 filters, a kernel size of 3, a stride of 2, with a relu activation function. The output of the second convolutional layer was then flattened and fed into a fully connected hidden layer with 100 neurons with a relu activation and then into another fully connected layer with a number of neurons equal to the number of classes, this layer utilized the softmax function.

The transfer learning CNN models each utilized their own individual base model but all had a similar head architecture of 100 neurons connected to 525 neurons which served as the classification layer. The transfer learning models chosen were MobileNet, ResNet50, ResNet101, and a CNN model trained on a subset of the dataset.

The transformer model created from scratch was based on the implementation proposed within the keras documentation based on the work of Dosovitskiy et al. (Salama 2021). This architecture consists of a patching layer which subdivides the image into 4x4 pixel patches that are then fed into two transformer layers that consist of a multi-head attention layer as described in the paper “Attention is all you need” (Vaswani et al.), a merge layer that combines the output of the attention layer with the patches, a normalization layer, and finally two fully connected layers each followed by dropout layers. After the transformer layers, the output is normalized, flattened, and then fed through three dropout layers each followed by a fully connected layer with the last layer being the classification layer. The transfer learning transformer model is based off of the ViT\_B32 model from “An image is worth 16x16 words” (Dosovitskiy et al. 2021). This base model was loaded without its head and a new one was constructed using a batch normalization layer, a fully connected layer of 100 neurons, another batch normalization layer, and a final fully connected layer of 525 neurons.

The models were trained for a number of epochs dependant on the class of the model; CNNs built from scratch were trained for 30 epochs, transfer learning CNNs were trained for 10 epochs, and vision transformer models were trained for five epochs. The number of epochs were chosen based on the training time such that each class had training times relatively within 30-90 minutes. While this resulted in the models not being fully trained it provided us with an indication of which models are improving the most and where further efforts should be directed.

**Experiment setting and Results:**

The dataset used for this training is the Birds 525 Species - Image Classification dataset from Kaggle (Gerry 2023). This is a dataset that is constantly being updated with new species and since the beginning of this project has been updated with an additional 10 species so that it now includes 525 species. All analysis, training, and testing was performed with the 525 species dataset. Presplit training, validation, and testing datasets were included; the validation and testing sets contained five images for each species that were handpicked as the “best” ones from the pool of possible images. The training images were the remaining images and had a varying number per species ranging from 130 images to 263 images. All images were pre-sized to be of size 224x224x3, the bird was the only species in frame and all images had good lighting. The subject of the images were captured in various poses ranging from standing on the ground, on a branch, in flight, or a close-up of the head.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Epochs | Training Time (minutes) | Training Accuracy | Validation Accuracy | Testing Accuracy |
| 2D 200 classes | 30 | 25 | 8.62% | 6.50% | 8.10% |
| 2D 525 classes | 30 | 60 | 12.65% | 11.88% | 11.73% |
| 3D 200 classes | 30 | 100 | 78.67% | 53.40% | 53.90% |
| 3D 525 classes | 30 | 250 | 49.15% | 43.69% | 46.49% |
| 3D 200 classes transfer 525 classes | 10 | 20 | 29.68% | 29.55% | 32.16% |
| MobileNet Transfer | 10 | 30 | 4.83% | 3.57% | 4.50% |
| ResNet50 Transfer | 10 | 55 | 62.94% | 66.87% | 68.62% |
| ResNet101 Transfer | 10 | 90 | 59.68% | 63.30% | 63.57% |
| ViT (Scratch) | 5 | 90 | 0.32% | 0.19% | 0.19% |
| ViT\_B32 (Transfer) | 5 | 40 | 11.02% | 11.22% | 12.00% |

*Table 1. Summary of training, validation, and testing results as well as training time*

The results of our model training and testing are presented in table 1. The CNN models trained from scratch had a large discrepancy between the 2D and 3D variations with the 3D architecture taking longer to train but producing more accurate results. The 3D 200 class model was able to achieve the highest training accuracy with 78.67%, however, its validation and testing accuracy were substantially lower (53.40% and 53.90% respectively).

Both the ResNet50 and ResNet101 models had excellent performance compared to the other models trained with testing accuracies of 68.62% and 63.57% respectively. While this does indicate that transfer learning methods have some benefit over training from scratch, appropriate model selection is important. MobileNet had the second worst performance with 4.50% testing accuracy.

The transformer model trained from scratch had the worst performance with 0.19% testing accuracy; given 525 classes this accuracy is equivalent to randomly guessing and shows no improvement after 90 minutes of training. Using the pretrained ViT\_B32 model as the base yielded much better results, achieving 12.00% accuracy after only 40 minutes of training.

**Discussion:**

The experiment began by training a 2D convolutional neural network (CNN) on a grayscale dataset, which took around 60 minutes for all 525 classes over 30 epochs. However, the results were not reasonable because there was little improvement in performance over that period. The training accuracy was 0.1265, the validation accuracy was 0.1188, and the test accuracy was 0.1173. The poor performance could be due to a lack of color in the dataset or too many classes at once while still trying to extract features.

To understand the problem better, we compared the performance of 2D CNNs and 3D CNNs as well as reducing the number of classes to 200. We found that 2D CNNs have the fastest time per epoch but very weak performance, and accuracy improvement is slow. On the other hand, 3D CNNs are slower to train than 2D CNNs but can learn the data better in the same number of epochs. Reducing the number of classes reduced the training time but the accuracy on the subset of data was similar to the full 525 class models. However, it is noted that the 3D 200 class model had far greater training accuracy than validation and testing accuracy. This is an indication that this model was overfitting the data and would benefit from a more complex model architecture, dropout layers, or additional training images in the form of image augmentation.

With this finding, we formulated a self-trained transfer model in which the classification head of the trained 3D 200 class model was replaced with a new 525 class classification head. This proved to be highly effective as the training time was the lowest of all the 525 class models and although the initial accuracy was mediocre compared to the ResNet transfer models there appears to be plenty of space for improvement in a relatively short amount of training time.

The other transfer learning models we tested had varying performance. The MobileNet and ViT\_B32 models had poor accuracy after training on the dataset and were surpassed by the ResNet50 and ResNet101 models. We hypothesize that the MobileNet architecture is too generalized and not complex enough for this classification task without significant training time. The ViT\_B32 model on the other hand may be too complex for the task and therefore isn’t able to learn as effectively. The ResNet models however, appear to lie in the sweet spot of architecture complexity so that they are able to distinguish between the species without overcomplicating the problem. This is evident in the fact that the 50 layer variant of ResNet performs slightly better than the 101 layer and hints that adding more complexity to the model may not be conducive to better performance.

Finally, the vision transformer model we built ourselves had poor performance while also taking a significant time to train. This was somewhat expected based on the findings of Dosovitskiy et al.’s work where they observed that transformer performance can directly be linked to the amount of data it was trained on. They show that sufficient dataset sizes are at least one million images and can range up to 300 million images, which is orders of magnitude larger than our training dataset which contains 82,724 images. For future work involving transformers, it appears necessary to use a pretrained model as the base as training one from scratch requires immense datasets and computational power to achieve results similar to those that are publicly available.

**Conclusion:**

**​​**The experiments presented in this paper show that the choice of deep learning model and training strategy can significantly affect the accuracy of image classification tasks. From the experiments, we can conclude that 3D deep learning models perform better than 2D models on bird image classification tasks, but they require more training time. Transfer learning approaches can improve the accuracy of deep learning models, especially when fine-tuning pre-trained models like ResNet50 and ResNet101. Moreover, the experiments highlight the importance of selecting the appropriate model for a given dataset. For example, the vision transformer model performed poorly when trained from scratch, but its accuracy improved significantly when fine-tuned on a pre-trained model. However, it still performed worse than the CNN variants. In summary, the results suggest that selecting the right model and using transfer learning approaches can significantly improve the accuracy of image classification tasks. However, achieving high accuracy requires careful selection of the model architecture, the amount of training data, and the training strategy.

**References**

Alharbi, F., Alharbi, A., & Kamioka, E. (2019). Animal species classification using machine learning techniques. *MATEC Web of Conferences*, *277*, 02033. <https://doi.org/10.1051/matecconf/201927702033>

Christin, S., Hervet, É., & Lecomte, N. (2019). Applications for deep learning in ecology. *Methods in Ecology and Evolution*, *10*(10), 1632–1644. <https://doi.org/10.1111/2041-210x.13256>

Dosovitskiy , A., Beyer , L., Kolesnikov , A., Weissenborn , D., Zhai , X., Unterthiner , T., & Dehghani , M. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Gerry. (2023, April 20). *Birds 525 species- image classification*. Kaggle. Retrieved April 2023, from <https://www.kaggle.com/datasets/gpiosenka/100-bird-species>

*Merlin Bird Id - home*. Merlin Bird ID - Free, instant bird identification help and guide for thousands of birds. (n.d.). Retrieved April 2023, from <https://merlin.allaboutbirds.org/>