Bird Species Image Classification

Akhil Krishna Nair
nair.akhi@northeastern.edu

Darsh Vora
vora.dar@northeastern.edu

Zeel Ashishbhai Jodhani
jodhani.z@northeastern.edu

***Abstract*— With the increasing threat of endangerment and extinction to many avian species, there has been a great need to conserve bird population across the planet. Identification of bird species plays an important role in the process of protecting bird species. Traditional approaches of identification of bird species are laborious and require great manual human efforts to visually identify the birds. There is a need for an efficient tool that scientists, government agencies, and the public can use to analyze amounts of bird data on a broad scale. With the help of Deep Neural Networks, we can create methods more efficient and accurate than human observationfor detecting bird species. With the help of Convolutional Neural Networks, which are a type ofDeep Neural Networks, we are able to classify the birds from their images to their respective species with a great amount of accuracy (90% on a test set).**

Keywords—Convolutional Neural Networks, Image Classification, Inception V3, VGG-16

I. INTRODUCTION

Traditional methods of bird species identification are not only laborious but limited to the ability of the individual human being performing the task. This process can be expensive and prone to error. Thus, there is a significant need for a more efficient technique to detect and identify bird species. There has been an increase in the number of bird species entering the endangered list and many from these end up extinct. There has been a great need for conservation of bird species. Birds are vulnerable to climate change and changes in the ecosystem as much as or even more than other animals. Lately, there have been various studies to investigate potential opportunities to tackle such threats to the avian population on our planet and various artificial intelligence techniques have recently gained popularity [1]. Our project aims to utilize Deep Neural Networks to classify images of birds to their respective species. The dataset available to use for this process consists of 89,885 images of birds labelled with their species names. This dataset [2] comes with a creative commons open-source license (Universal Public Domain Dedication) ensuring that it can be freely used, modified, and distributed, enhancing its accessibility, and allowing for data driven studies to be performed with the same. The images are 224x224x3, meaning they are RGB (3 channels), and the birds cover 50% or more of the images. We will use these images to train a Convolutional Neural Network to learn the various features of these birds and associate these to their species.

II. BACKGROUND

Convolutional Neural Networks have been popular in the domain of image classification and are a great option for our use case. These networks are especially good at detecting features from images and making predictions on the same. Deep learning has been used by other individuals to perform image classification for bird images in prior work and they have achieved great results. Inception V3 and VGG-16 are some models trained on the ImageNet dataset and have been very commonly used for image classification and have been proved to be useful to classifying animals before. Prior work in this domain includes transfer learning applied on these models to fine-tune them to specific datasets containing bird images. Transfer learning has been useful in cases where there have been less data available. In such cases, training a network from scratch would be difficult and transfer learning applied to a robust network like Inception V3, Efficient Net, VGG-16, etc. would provide great results.

Study [3] utilizes Google's Inception-v3 algorithm within the TensorFlow framework to classify animals using machine learning approaches. The model is retrained on datasets of mammals using transfer learning, and it exhibits a significant improvement in classification accuracy, up to almost 95%.

[4] uses a weighted average transfer learning approach which combines the results from various models like EfficientNetB5, VGG19, Inception V3, ResNet152, to make inferences. Transfer learning is used to train the model on the dataset and they achieve great results.

Our project aims to compare the results obtained from training a Convolutional Neural Network built from scratch which draws inspiration from the VGG-16 architecture. Our network uses fewer layers than the VGG-16 network which consists of over 35 million trainable parameters. Our model contains 3.5 million parameters. We also attempt to rely on transfer learning applied to Inception V3 and Efficient Net models.

III. Approach

A convolutional neural network consists of convolutional layers and pooling layers, followed by a fully connected layer at the top (the end) of the network. A convolutional layer contains filters (also known as kernels) which are convolved with the input data (which is usually 2 or 3 dimensional) to generate feature maps.

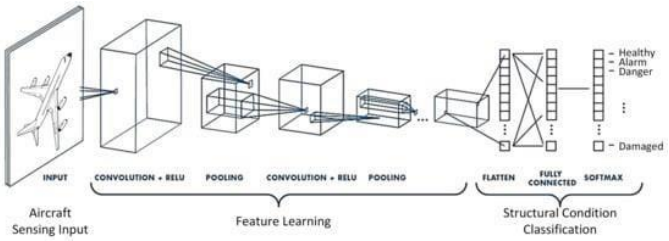

Fig. Convolutional Neural Network

Pooling layers are used to reduce the dimensionality of the inputs. Pooling performed with a pool of size 2 on an input of shape 200x200 would result in a 100x100 output.

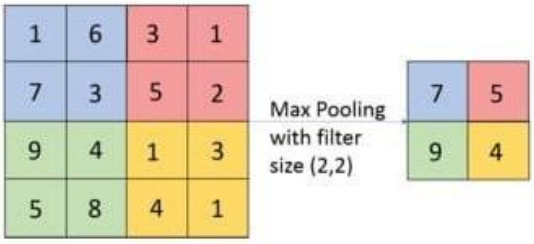

Fig. Result after applying max pooling with a 2x2 filter

We process our input images with the help of the TensorFlow Keras library and its ImageDataGenerator class which can read and process images to train the model. In practice, we want the model to learn on variations of the actual training input since in the real world, inputs for inference may be variations of the actual training. These variations include brightness, contrast, rotations, shearing, shifts in width or height, and so on. We call these transformations as augmentations and apply these to our input data before passing them to the model for training. These transformations will help our model learn to predict in more variant conditions than the original images intend to. This step also helps us create more training data.

The augmentations we use in our dataset are:

1. Rotation
2. Horizontal Flipping
3. Brightness
4. Shift across width and Height
5. Scaling down image values
6. Zoom

Fig. Data Augmentations Visualized

Next, we define the model architecture to be used. This step requires us to calculate the effect of having various convolutional and pooling layers based on the input size of the images. This step will also dictate the number of parameters to expect in our model. A table like shown below is a good way to demonstrate the model architecture.

The first architecture we trained is shown below.

Layer (type)	Output Shape	Param #
conv2d_62 (Conv2D)	(None, 224, 224, 16)	448
max_pooling2d_62 (MaxPooling2D)	(None, 112, 112, 16)	0
batch_normalization_1 (BatchNormalization)	(None, 112, 112, 16)	64
conv2d_63 (Conv2D)	(None, 112, 112, 32)	4,640
max_pooling2d_63 (MaxPooling2D)	(None, 56, 56, 32)	0
batch_normalization_2 (BatchNormalization)	(None, 56, 56, 32)	128
conv2d_64 (Conv2D)	(None, 56, 56, 64)	18,496
max_pooling2d_64 (MaxPooling2D)	(None, 28, 28, 64)	0
batch_normalization_3 (BatchNormalization)	(None, 28, 28, 64)	256
conv2d_65 (Conv2D)	(None, 28, 28, 128)	204,928
max_pooling2d_65 (MaxPooling2D)	(None, 10, 10, 128)	0
batch_normalization_4 (BatchNormalization)	(None, 10, 10, 128)	512
conv2d_66 (Conv2D)	(None, 5, 5, 256)	295,168
max_pooling2d_66 (MaxPooling2D)	(None, 2, 2, 256)	0
batch_normalization_5 (BatchNormalization)	(None, 2, 2, 256)	1,024
flatten_13 (Flatten)	(None, 1024)	0
dense_26 (Dense)	(None, 32)	32,800
batch_normalization_6 (BatchNormalization)	(None, 32)	128
dense_27 (Dense)	(None, 526)	17,358

Fig. Shallow CNN Architecture

This architecture had a total of 575950 parameters. The batch normalization layers helped to keep the values at the end

of various layers normalized. This helps speed up training. This model achieved a validation accuracy of 77% which was good but had scope to be improved. The below diagram shows the training steps and accuracies during training. The augmentation of training images and the model not being too deep could be the reason behind the training accuracy being lower than the validation accuracy. But this indicates that we are not overfitting on the training data, which is possible with a model too complex for the problem, or with too less training data.

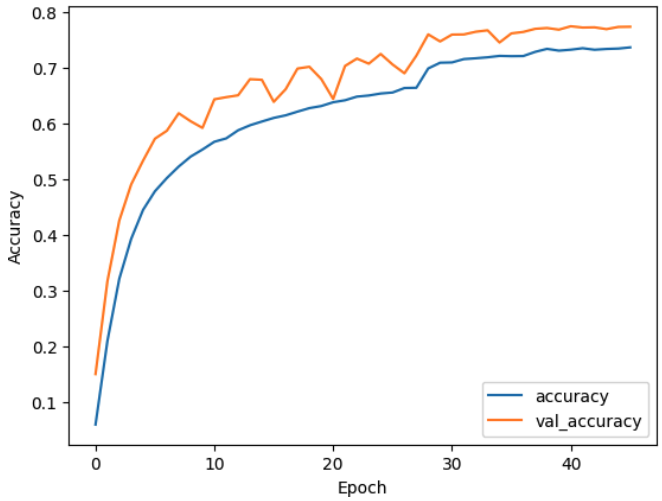

Fig. Training vs Validation accuracy

Drawing inspiration from the VGG-16 architecture, we restructure our model architecture to extract more features from the images and further hope to improve our accuracy.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
batch_normalization (BatchNormalization)	(None, 112, 112, 32)	128
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18,496
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 56, 56, 64)	256
conv2d_4 (Conv2D)	(None, 56, 56, 128)	73,856
conv2d_5 (Conv2D)	(None, 28, 28, 128)	409,728
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 10, 10, 128)	512
conv2d_6 (Conv2D)	(None, 10, 10, 256)	819,456
conv2d_7 (Conv2D)	(None, 5, 5, 256)	1,638,656
max_pooling2d_3 (MaxPooling2D)	(None, 2, 2, 256)	0
batch_normalization_3 (BatchNormalization)	(None, 2, 2, 256)	1,024
dropout (Dropout)	(None, 2, 2, 256)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 525)	538,125

Fig. Deep CNN Architecture

Total params: 3,547,309 (13.53 MB)

Trainable params: 3,546,349 (13.53 MB)

Non-trainable params: 960 (3.75 KB)

This model had significantly larger number of parameters and went on to take longer to train. We did observe an improved validation accuracy score of 85%.

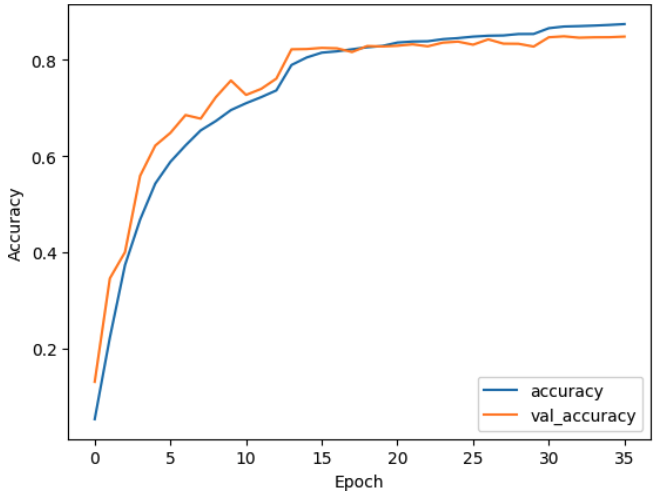

Fig. Training vs Validation accuracy

We also use InceptionV3 model to perform transfer learning using our dataset. In order to do this we, freeze the weights of the model while adding our fully connected layer at the top of the layer and retraining with our dataset. While the results from this method did not reach the same level as our Convolutional Neural Network trained from scratch, it was noteworthy as training through transfer learning was much faster than training the CNN from scratch.

IV. Results

The best results were observed with the deep CNN trained from scratch as it obtained an accuracy score of 90.5% on the test set as opposed to 87.5% obtained by the transfer learning on InceptionV3 model. The CNN with less layers was able to obtain an accuracy score of 78% on the test set.

Fig. Predicted values for birds vs ground truth

On the test set, the deep CNN achieved an F1 score of 90.4%, whereas the InceptionV3 model trained with transfer learning achieved an F1 score of 85.5%.Precision, Recall, F1 scores for each class was observed and the claimed overall scores were weighted averages of all the classes together. The below figure shows the individual class scores for various birds.

Fig. Precision, Recall and F1 scores for classes (truncated version)

V. Discussion

Our test set compromised of 5250 images which were never observed by the model training. Since the model was able to classify the test images with such high accuracy, we can say that this model can be used with high confidence to identify bird species. While we did not achieve higher scores with the transfer learning approaches, there is further scope for improvement through better hyperparameter tuning methods to train the model efficiently. Unfreezing more layers from the InceptionV3 and VGG16 models would also help us fine tune the model to our dataset better and achieve better scores than the deep CNN model that we used in our project. Since our training accuracy and validation accuracy are very close to each other, there might be scope to increase model complexity and achieve a higher training and validation score as observed from moving from the original CNN to the deeper CNN with added convolutional layers.

A higher validation and test accuracy than training accuracy could imply that our validation and test sets are simple and very similar to the train dataset, which is to be expected in this case. As we applied augmentation steps only to the training dataset and the model was trained on variations in brightness, zoom, rotations of birds in the images, it is expected to have a lower training accuracy when predicting classes for validation and test sets which are without any such transformations. We used dropout layers which randomly invalidated the outputs of certain units in the network in order to create a regularization effect during training which also explains the lower training accuracy as dropout is not applied during validation and testing of the model.

VI. Conclusion

Our project was able to demonstrate the abilities of deep Convolutional Neural Networks in the domain of image classification. An accuracy of 90% would be very useful in implementing tools to aid in studies pertaining to the conservation of bird species. This implies huge savings in terms of time and money since these models might be able to surpass humans in terms of accuracy and speed.

	precision	recall	f1-score
ABBOTTS BABBLER	0.87	0.76	0.81
ABBOTTS BOOBY	0.71	0.43	0.54
ABYSSINIAN GROUND HORNBILL	0.82	0.82	0.82
AFRICAN CROWNED CRANE	0.93	1.00	0.96
AFRICAN EMERALD CUCKOO	0.96	0.72	0.83
AFRICAN FIREFINCH	0.87	0.62	0.73
AFRICAN OYSTER CATCHER	1.00	1.00	1.00
AFRICAN PIED HORNBILL	0.70	0.81	0.75
AFRICAN PYGMY GOOSE	0.97	0.94	0.96
ALBATROSS	0.49	0.76	0.59
ALBERTS TOWHEE	0.87	0.92	0.89
ALEXANDRINE PARAKEET	0.92	0.92	0.92
ALPINE CHOUGH	0.89	0.97	0.93
ALTAMIRA YELLOWTHROAT	0.84	0.84	0.84
AMERICAN AVOCET	1.00	0.93	0.96
AMERICAN BITTERN	1.00	0.93	0.97
AMERICAN COOT	0.94	0.86	0.90

Fig. Performance Evaluation Metrics

References

[1] B.P. Toth and B. Czeba, "Convolutional Neural Networks for Large-Scale Bird Song Classification in Noisy Environment", CLEF (Working Notes), pp. 560-568, 2016, September.

[2] https://www.kaggle.com/datasets/gpiosenka/100-bird-species

[3] Bankar Jyotsna and Nitin R. Gavai, "Convolutional neural network based inception V3 model for animal classification", International Jour-nal of Advanced Research in Computer and Communication Engineering, vol. 7, no. 5, pp. 142-146, 2018.

[4] V. R. Murthy Polisetty and S. Chokkalingam, "Enhancing Bird Species Classification: A Weighted Average Transfer Learning Ensemble Approach," 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2024, pp. 1401-1406, doi: 10.1109/IDCIoT59759.2024.10467506. keywords: {Deep learning;Biological system modeling;Transfer learning;Training data;Predictive models;Birds;Data models;Image Classification;Bird Species Classification;Transfer Learning;Data Augmentation;Ensemble Prediction;Computer Vision}