SPRINGBOARD DSC CAPSTONE PROJECT 2

CLASSIFYING IMAGES OF BIRDS

USING THE 2019 iNATURALIST DATASET

by: Lauren Broussard

OUTLINE

- PROBLEM & POTENTIAL CLIENT(S)
- APPROACH AND DATA SET
- DATA WRANGLING
- EXPLORATORY DATA ANALYSIS
- IN-DEPTH ANALYSIS
- RESULTS
- RECOMMENDATIONS
- FURTHER CONSIDERATIONS

Image: Canva.com

PROBLEM & POTENTIAL CLIENT(S)

CAN WE USE DEEP LEARNING TO DISTINGUISH IMAGES OF BIRDS FROM OTHER WILDLIFE TYPES?

> Image classification of the natural world has many benefits
> Can help track migratory patterns without disturbing habitats
> Can be used for website tags and accessibility

POTENTIAL CLIENTS

Wildlife Researchers

Bird Watchers

Social Media Managers

APPROACH & DATASET

> DATA SOURCED FROM INATURALIST/KAGGLE

- This project uses jpeg images and json files comprising 1010 different species and 6 wildlife categories from the iNaturalist 2019 Kaggle competition. The images are taken from a variety of angles and in a variety of environments.

> DOWNLOADED, CLEANED, & PREPROCESSED

- The original, pre-cleaned dataset consisted of 265,213 images, with 18% (47,867) of images consisting of birds. A subset of the images were randomly chosen, and the dataset was balanced to get a 50/50 split of birds and non-birds for training and testing.

> DEEP LEARNING/CNN FOR IMAGE CLASSIFICATION

- Two models were created using deep learning - one simple neural network, and another using convolutional neural networks - to classify whether or not there was a bird in the presented image. The keras library in Python was used for this analysis.

Image: Canva.com

Image: Canva.com

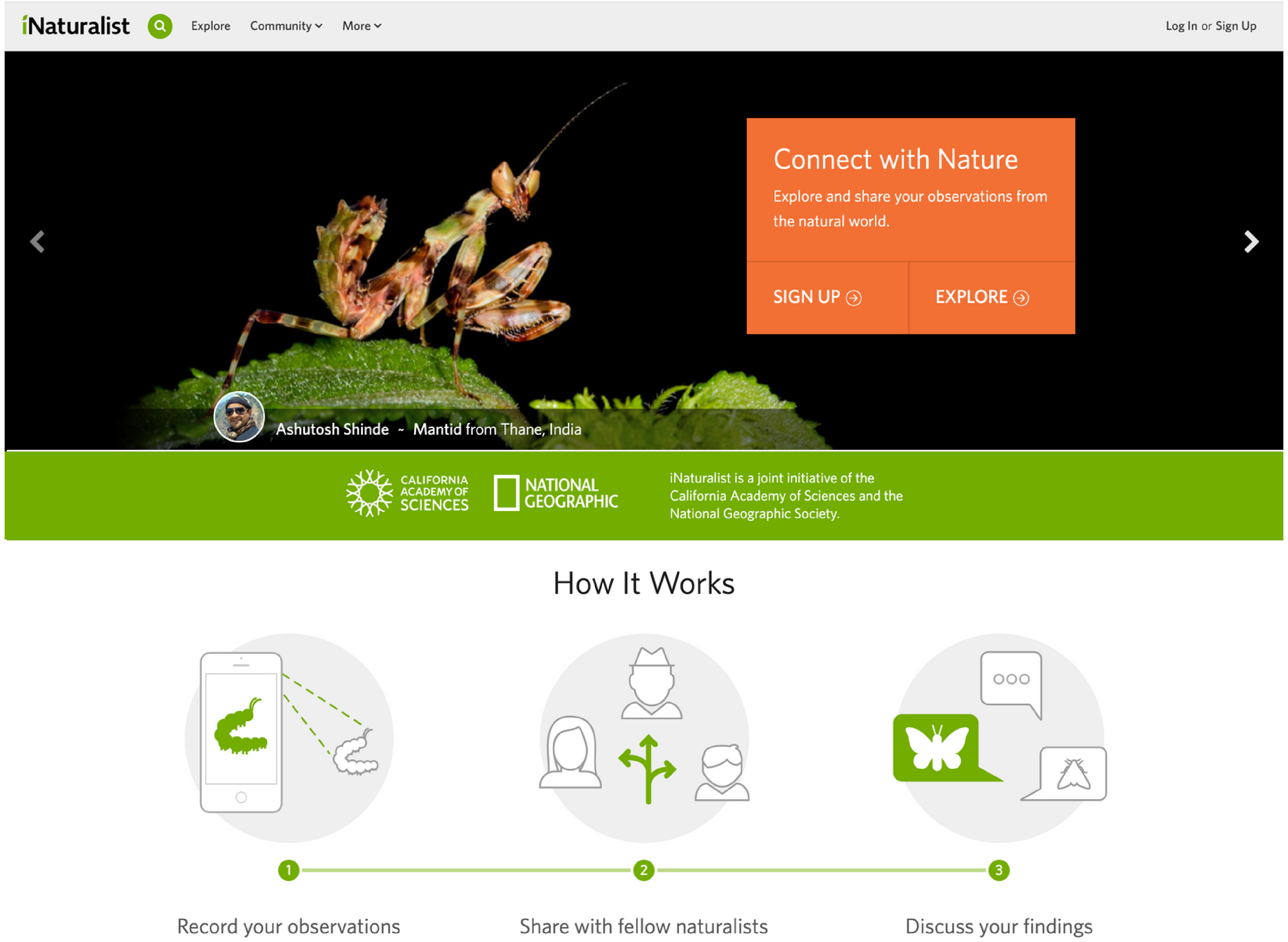


Image: inaturalist.org

ABOUT INATURALIST

iNaturalist allows users to add images of wildlife they observe around the globe. They host an annual Kaggle competition that encourages participants to help improve classification for their repository of images.

Unless otherwise noted, all images are from iNaturalist's provided dataset.

DATA WRANGLING

The downloaded data was already fairly cleaned and standardized. However, the following steps were taken to prepare the data for modeling:

Loading Data

The download from Kaggle included both annotations and images. Annotations used for this project came in the form of two json files (57MB and 816.46KB respectively). DataFrames were created for each of the files and were eventually merged together. Columns included filename, image width, and image height. The images themselves were stored in folders which included their main category name (i.e. Plants, Birds, Reptiles, etc.) and were further divided into subfolders by species. The image folders included a total of 265,213 images (79GB).

Adding/Removing Columns

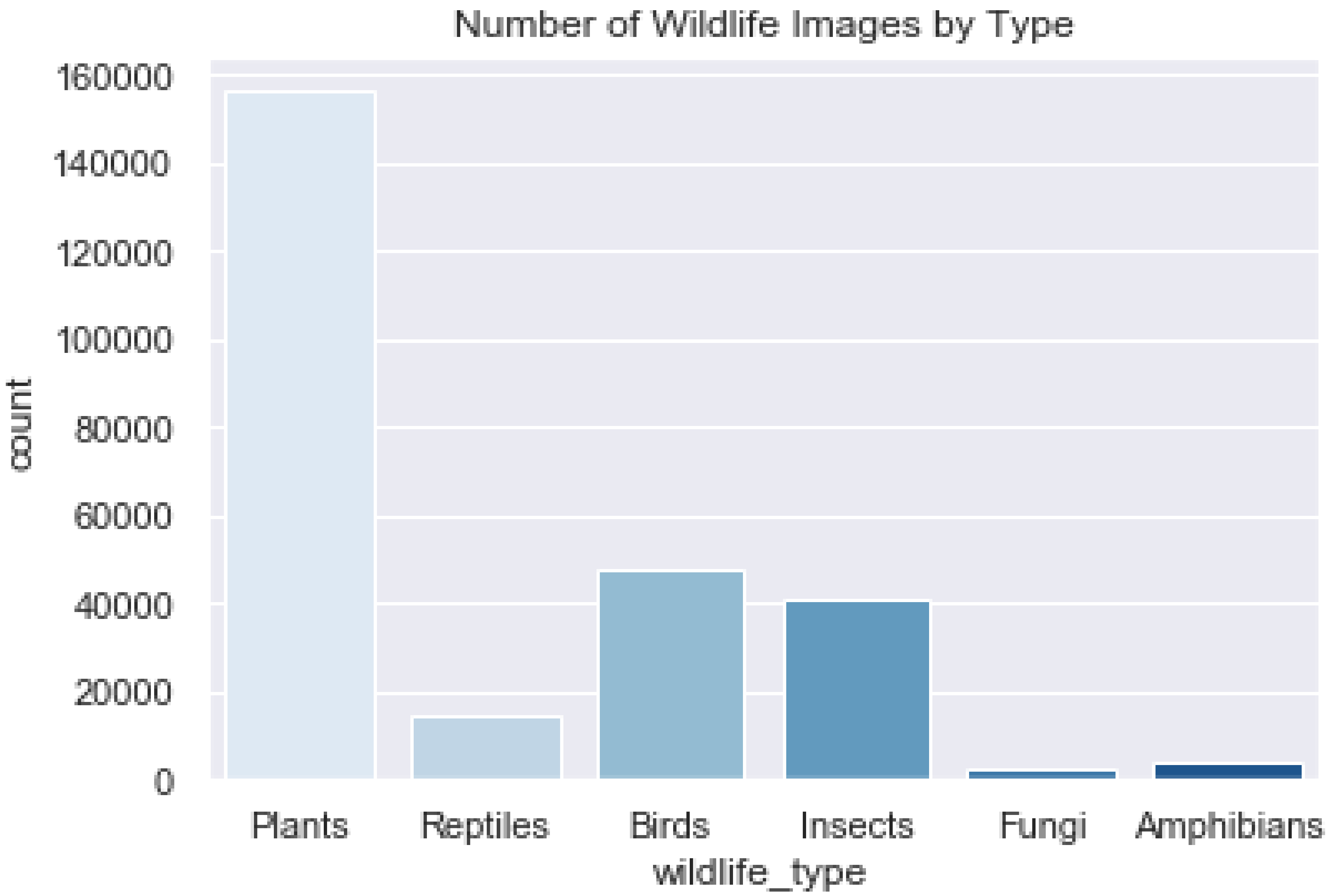
Information was pulled from the 'annotations' and 'images' keys and put into a dataframe. To be more descriptive, the column ‘category_id’ was changed to ‘species_id’. Additionally, since we were interested in labeling Birds, we pulled the ‘iconic category’ name from the subfolders in the ‘file_name’ field. This created field was then named “wildlife_type.”

Numerical Encoding

The "wildlife_type" column was used to create another binary column called "is_bird," that gives a 1 if the ‘wildlife_type’ is Birds, and 0 if it is not, This field was used as the target value for the classification problem.

EXPLORATORY DATA ANALYSIS (EDA)

The vast majority of images in the dataset (nearly 160,000) are of Plants, followed by Birds and then Insects. **The dataset included 47,867 images of birds, or 18% of all images.** There is a class imbalance, which could affect our model -- this was remedied later in the project by randomly selecting equal samples of birds vs all other categories.








EDA:
BIRD IMAGES
FULL DATASET

47,867
Number of Images

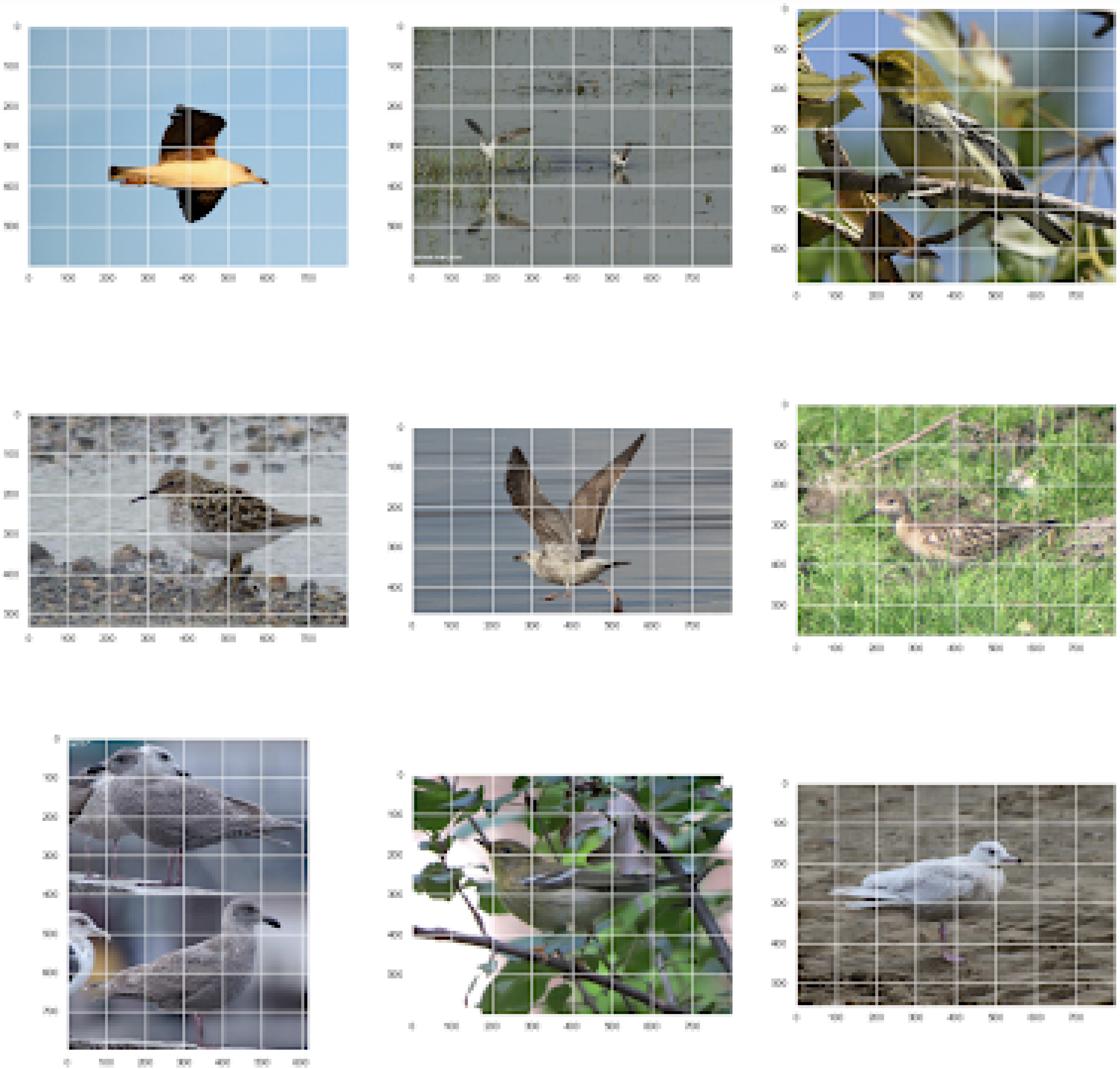
126
Number of Species

130/188
Min. Pixel Height/Width

800/800
Max Pixel Height/Width

EDA:
BIRD IMAGES

The images consisted of wildlife mainly in their natural environments - in grass, in the air, in trees, etc. The images were also taken in different orientations and at different angles. This required a model that was able to cut through the background noise images.



IN-DEPTH ANALYSIS

SUPERVISED LEARNING

APPROACH:
DEEP LEARNING &
CONVOLUTIONAL NEURAL NETWORKS (CNN)

> STEP: PRE-PROCESSING

- Balance Dataset: 50/50 Random Split of Birds and Non-Bird Images
- Split Dataset: Train/Validation/Testing
- Augment Images & Set Up Training/Validation Generators

> STEP: RUN INITIAL MODEL: MODEL 1

- **Simple Neural Network,** 5 hidden layers
- 5,000 Images | 25 Epochs
- Prediction Accuracy = 0.49

> STEP: UPDATE ARCHITECTURE/ PARAMETERS

- Increase: Epochs, # of Images, Batch Size
- Add: Cov2D and MaxPooling Layers

> STEP: RUN NEW MODEL: MODEL 2

PRE-PROCESSING & SET-UP

ARCHITECTURE: MODEL 2

COV2D (4): Since pixels are typically correlated with their neighbors. Looks for correlation between edges and contours.

MAX POOLING LAYERS (4): Helps with a large number of parameters, as is the case with the original model. It summarizes groups of pixels based on their max value.

DROPOUT (1): The dropout layer randomly sets inputs to 0 at a rate of 0.6 at each step during training, which according to the documentation can help prevent overfitting.

DENSE (1): Every node/pixel is connected to all the units of the previous layer.

PRE-PROCESSING & SET-UP

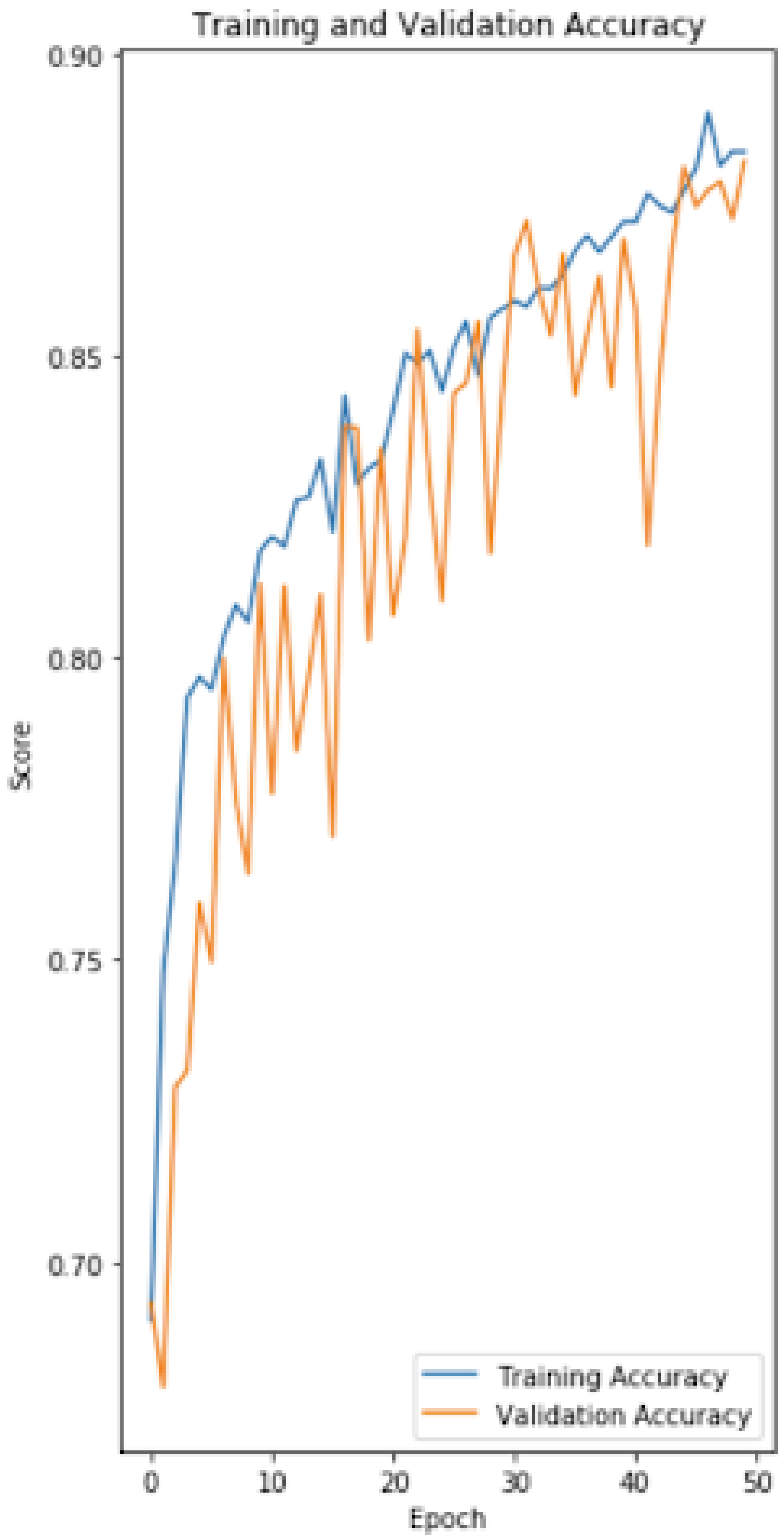
ARCHITECTURE: MODEL 2

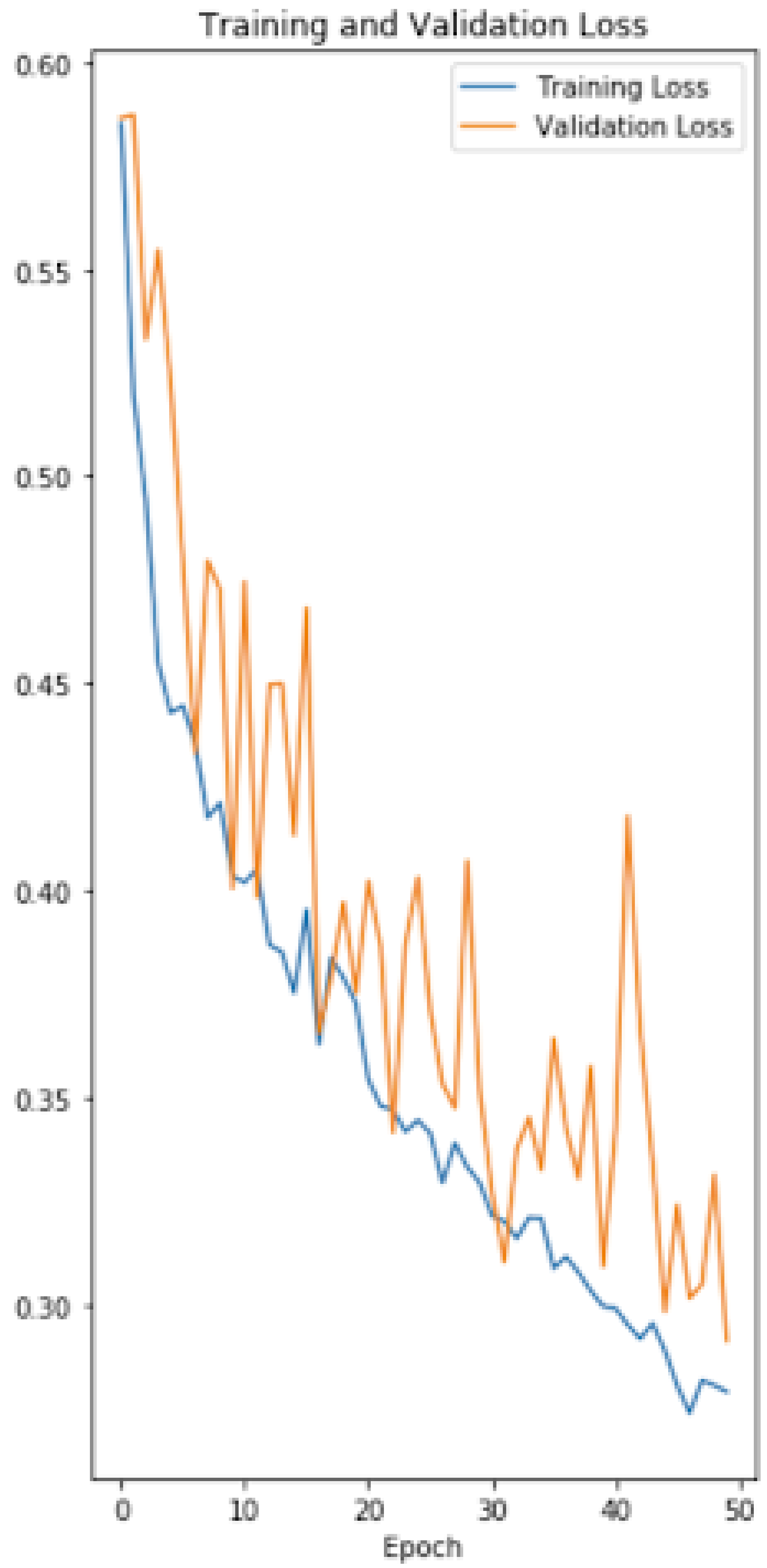
```
model2 = Sequential()

model2.add(Conv2D(32, (3, 3), input_shape=(img_width,img_height,3)))
model2.add(Activation('relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Conv2D(32, (3, 3)))
model2.add(Activation('relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Conv2D(32, (3, 3)))
model2.add(Activation('relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Conv2D(64, (3, 3)))
model2.add(Activation('relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Flatten())
model2.add(Dense(100))
model2.add(Activation('relu'))
model2.add(Dropout(0.6))
model2.add(Dense(1))
model2.add(Activation('sigmoid'))
```

FINAL MODEL PARAMETERS

- **Total # of Images: 15,000**
  - The addition of more training images helps the model better learn what an image of a bird looks like.
- **Increased Epochs: 50**
  - The number of epochs increased from 25 to 50 for longer training.
- **Callbacks: Early Stopping Monitor**
  - The early stopping monitor was increased from 2 to 10 to allow the model more time for learning.
- **Prediction Batch Size: From 128 to 1**
  - Allowed the model to look at one image at a time for predictions.
- **Total Training Parameters: 268,489**
  - Decreased between models from original 16,782,889.

IN-DEPTH ANALYSIS





TRAINING & VALIDATION

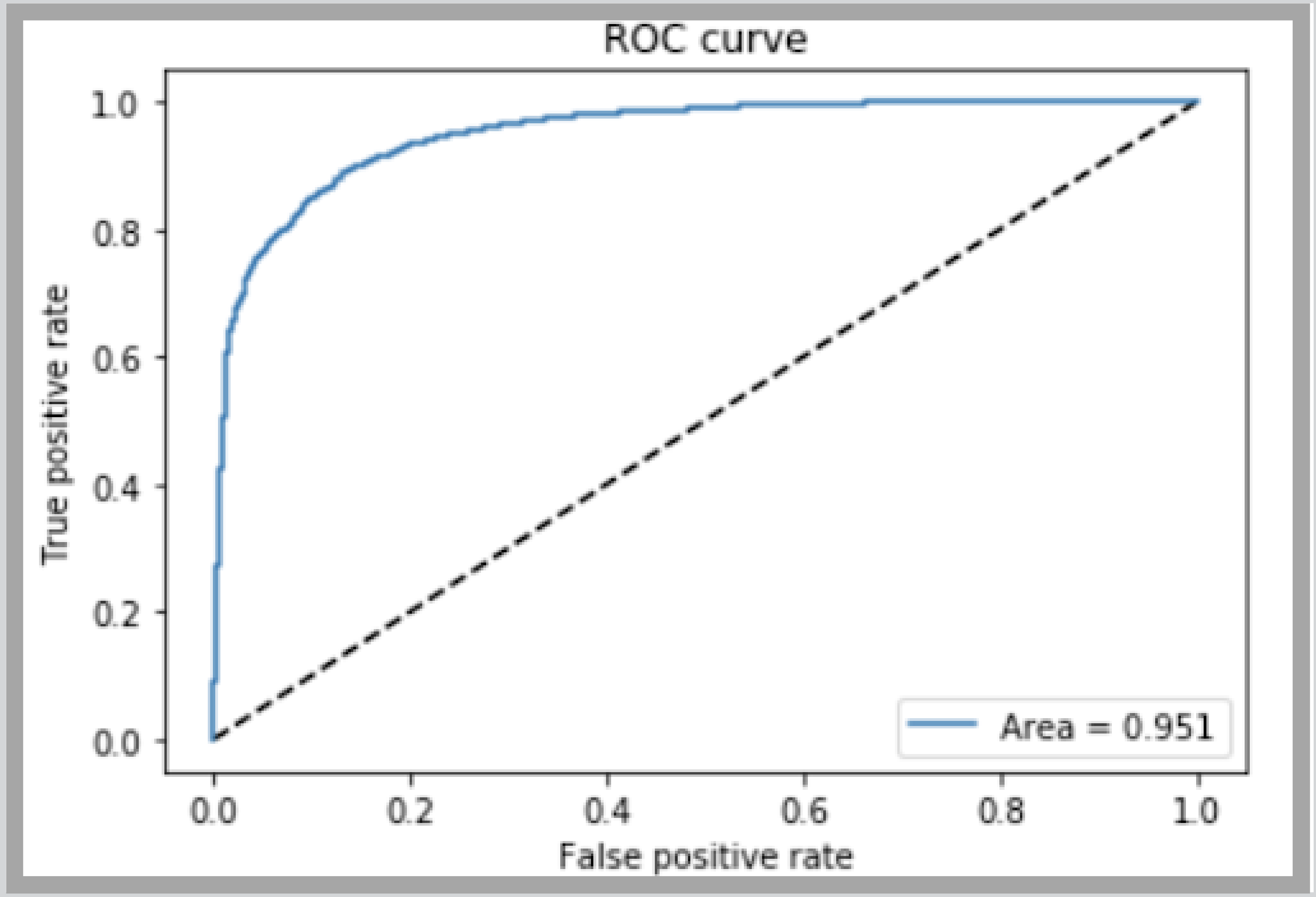
Running the generator through 50 epochs of training and validation, **the model achieved its highest validation accuracy at 88%**. The training seemed to increase the accuracy at a fairly steady pace, though the validation accuracy seemed to bounce around between 80% and 87%.

RESULTS

CLASSIFICATION REPORT & ROC CURVE

Classification Report

	precision	recall	f1-score	support
Not Bird	0.87	0.88	0.88	1830
Bird	0.88	0.86	0.87	1770
accuracy			0.87	3600
macro avg	0.87	0.87	0.87	3600
weighted avg	0.87	0.87	0.87	3600



The model appeared to be able to identify images where there was no bird slightly better than it was able to predict images where there was a bird. However, **the model was still able to classifying birds well, with an F1 score of 87%.**

RESULTS

EXAMPLE PREDICTIONS: CORRECT (87%)

BIRD

Probability of Bird: 0.999685

NOT BIRD

Probability of Bird: 0.111596

BIRD

Probability of Bird: 0.931944

The model did well at classifying birds, especially since many of the birds were photographed in their natural habitats, and were sometimes obscured. This is promising for automatic classification where it is necessary to not disturb an animal's habitat.

RESULTS

EXAMPLE PREDICTIONS: INCORRECT - MISIDENTIFIED AS BIRD

ACTUAL: NOT BIRD (AMPHIBIAN)

Probability of Bird: 0.999685

ACTUAL: NOT BIRD (INSECT)

Probability of Bird: 0.990383

ACTUAL: NOT BIRD (PLANT)

Probability of Bird: 0.997390

The model incorrectly classified only 13% of the test images presented to it. The images above are examples where the model was confident that the image was of a bird, although it was an image of something else.

RESULTS

EXAMPLE PREDICTIONS: INCORRECT - FAILED TO ID BIRD

PREDICTED: NOT BIRD

Probability of Bird: 0.075025

PREDICTED: NOT BIRD

Probability of Bird: 0.135277

PREDICTED: NOT BIRD

Probability of Bird: 0.098906

When the model made an error in classification, it was more often that the model did not find a bird, when there was in fact one there.

RECOMMENDATIONS

& FUTURE CONSIDERATIONS

Use Model to Shorten Classification Work

Since the model predicts well, it can be a good starting point in classifying birds. For instance, a bird watcher would have a starting place to begin looking.

More Training Images

More training images will allow the model to learn more features of birds. This will require increased computing power, which was not available for this project.

Classify By Species

Identification by individual species can help track migration patterns of particular birds, or identify when endangered species are brought to a habitat they should not be in.

Different Model

Future iterations of the project could also look to other pre-trained models, such as VGG16, which is a CNN model trained on the ImageNet database.

THANK YOU

LAUREN BROUSSARD

linkedin.com/in/laurenbroussard