# Problem Statement

The primary objective of our project is to train a CNN model that can accurately classify bird species based on the visual features present in the images. The model will need to successfully differentiate between 20 species of birds.

# Problem Set Up

The 20 bird species used for this project are a random selection of the 525 species in this [Kaggle dataset](https://www.kaggle.com/datasets/gpiosenka/100-bird-species), but include a variety of birds. They include larger birds such as herons and eagles as well as smaller birds such as barbets and quails.

The training data includes 100 images per species. The data author also provided a validation and a test with 5 images of each species in both. The images are all size 224x224 and contain only a single bird that comprises at least 50% of the pixels in the image.

The training dataset contains more male birds than female birds, giving it a bias towards the males, which tend to be more colorful and decorative.

ADD SOME BIRD PICS

# Model Implementation

## Baseline Pretrained Model

As a benchmark for the problem, an EfficeintNetB0 pretrained model was used to set a baseline. Using ImageNet weights, the model achieved an accuracy of 100% on the validation data after only 4 epochs. This indicated that this problem was sell suited to a neural network architecture.

TABLE FOR EFFICIENT NET

## Initial Model

To evaluate a starting point for the model, and assess the classes that were performing well, an initial model was trained with 1 CNN layer, a pooling layer, and a softmax prediction layer.

Using early stopping, the model achieved a validation accuracy of 81% within 21 epochs.

INSERT TABLE OF RESULTS

This initial model was also showing some signs of overfitting with validation accuracy starting to decrease in the later epochs, although the early stopping prevented any severe overfitting.

PLOT OF EPOCHS

Reviewing the classes that were predicted well compared to the classes that were predicted poorly, a few patterns jumped out.

Birds with very distinct patches of color or distinctive feathering were being classified very well. Birds with muted colors or a lot of variation between images were being classified very poorly.

PICTURES OF WELL CLASSIFIED AND POORLY CALSSIFIED

TABLE WITH RESULTS

This indicated that the model was picking up the very prominent features, but wasn’t identifying more nuanced features needed for the less distinctive birds.

# Model Refinement

## Increased Complexity

To pull out more features from the images, we increased the complexity of the model by adding additional convolutional layers, and to support model invariance we included pooling after each layer.

DETAILS OF MODEL

Training the more complex model resulted in mixed results. While some species had higher levels of classification, other species had significant decreases in their classifications.

The pattern seen in the initial model continued with the more complex model and the birds that were classified better tended to be colorful, distinctive birds. While the ones that did not improve or got worse were less distinctive.

TABLE/IMAGE OF BIRDS THAT IMPROVED OR GOT WORSE

The overall result of these shifts was that the model’s accuracy decreased, and the loss increased. The model also showed signs of overfitting with the loss increasing as runs continued.

SHOW PLOT OF LOSS/ACCURACY

## Grayscale Image Evaluation

To validate the assumption that the model was relying heavily on color, we retrained the more complex model but used gray scale images instead of color.

Evaluating these results showed a clear trend that very colorful birds that had been well classified in the previous model, were now being poorly classified. However, birds with more distinctive feathering that had been classified well previously were still being classified well.

SHOW PICTURES OF BIRDS

## Progressive Resizing

In an effort to force the model to look at more general features, we employed progressive resizing ADD ARTICLE LINK to train the model. Because our original images were relatively high quality, the model was able to easily identify the highly distinct features for a subset of species, and was focusing on those to make its predictions while not learning about birds that were less distinct or had a lot of variation in color and position. To help the model focus in on other features that were less distinct we trained 3 separate models.

### 56x56

The base of our progressive resizing model was trained on 56x56 images. We started with the complex model built in the previous step and made adjustments to account for the changes.

Because the images were lower quality, we increased the filter size to 128, 64, 32 and on the three layers in order to draw out more features. Because the images were less distinct we were not as worried about the model learning the noise. We also included an additional Dense layer before the softmax layer to introduce additional complexity and differentiation to the model.

To help prevent overfitting, we included a dropout layer after the final MaxPooling layer with a 20% dropout rate. We then removed early stopping and allowed each model to run for 100 epochs.

This model performed extremely well before even adding in the higher quality images. The accuracy jumped to 88.99% and several birds that had been poorly classified on the complex model, such as the Bald Eagle and Bald Ibis were being 100% correctly classified.

BIRD PICTURES

Loss/Accuracy plot? Or just show that for 224?

### 112x112

An additional CNN layer with input size of 112x112 was added to the model. To avoid picking up too much noise from the higher quality images, filter size was reduced to 16 for this new layer. A MaxPooling layer was used to reduce the size to 56x56.

This was then added to the 56x56 model and the new layer was trained while locking the weights of the previous layer to retain the more general learning of those layers.

This model improved the accuracy again, and most species were either better classified or the same.

BIRD PICTURES? PLOT/LOSS CHART?

### 224x224

The final layer was another convolutional layer with a filter size of 16 and an input size of 224x224.

## Additional Images

# Test Results

# Next Steps