Identifying the Bird Species by their Call Using Convolutional Neural Network

Karthika Selvaraj

May 15th, 2023

Abstract

Researchers can gain a better understanding of bird populations, behavior, and the effects of environmental conditions on their survival by classifying the many bird species based on their vocalizations. Convolutional neural networks and a deep learning technique are used in this study to identify different bird species based on their calls. The Xeno-Canto website, which collects animal sounds from all around the world, is where the bird cries were collected from. 600 audio recordings of 12 different bird species were chosen for this study. It briefly describes the many procedures needed in creating a model, from audio feature and pattern extraction through processing a picture as an input to create an image classifier. The study also looks at how various options for network structure and hyperparameters can affect the model's quality.

1 Introduction and Overview

Environmental preservation and sustainable development have become more well known in recent years. The demand for ML models to distinguish the different bird species from their visual and audio inputs has increased as a result of birds' significance to ecology. In contrast to video-based monitoring, where images can be easily distorted, sound has the benefit of being able to travel over a great distance without being impeded by anything between the emitting source and the recording devices. A 343 (time) x 256 (frequency) "image" of the bird call is created by preprocessing the audio inputs to create spectrograms for each 2-second window. Convolutional neural networks (CNN) model, which is created especially for image recognition, is given these photos as input. They minimize the dimensionality of the feature maps and employ various layers to identify features in the input image. Because of this, they excel at tasks like image classification.

2 Theoretical Background

2.1 Neural Network

Neural networks are a specific sort of machine learning model that have been adapted from the structure and function of the human brain. They consist of networked nodes or neurons that process data coming in and produce predictions as an output. A single layer of neurons makes

up the simplest type of neural network, known as a single-layer network, which processes input data using a set of weights and biases to produce an output that is a prediction. While multi-layer neural networks feature two or more layers of neurons to recognize more complex patterns in the input data, this type of neural network is mostly used for binary classification. The input layer is the top layer, the output layer is the bottom layer, and the concealed layers are the middle layers. The input from the layer below is received by each neuron in a hidden layer, which then generates an output that is transferred to the layer above. The network is able to understand complex patterns in the data and is better suited for tasks that demand predictions with greater complexity thanks to superior performance on difficult tasks like speech or image recognition.

2.2 Convolutional Neural Network (CNN):

By identifying unique features or patterns everywhere in the image that define each distinct object class, convolutional neural networks (CNNs) have grown to somewhat mirror how humans classify images. Convolution layers and pooling layers, two specialized subtypes of hidden layers, are combined.Through convolution layers, the network first recognizes low-level elements in the input image, such as minute edges and color patches. Then, through pooling layers, these low-level features are pooled to create higher-level features. In the end, whether these higher-level traits are present or absent affects the likelihood of a specific output class. The CNN layer is nothing more than the numerous convolution and pooling layers used in conventional neural network topologies.

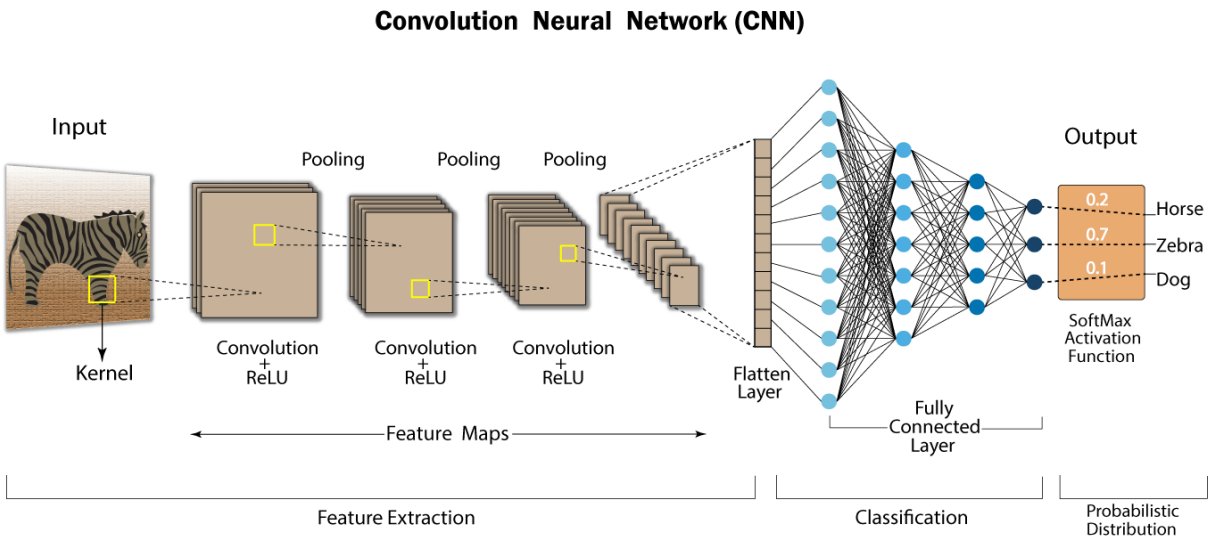
2.2.1 Architecture of Neural Network:

The foundational component of the CNN is the convolution layer. It carries the majority of the computational load on the network. A convolutional layer of a convolutional neural network (CNN) applies a group of filters or kernels to the input data. By sliding the filter over the input data for each filter and computing the dot product between the input values at each location and the accompanying filter weights, the input data gets convoluted. A feature map, the result of the convolution process, highlights different patterns or features in the input data. Each filter can learn to distinguish different characteristics, such as edges, corners, or textures, by altering its weights during the course of the training process.The number of filters and their sizes can be changed depending on the characteristics of the input data and the desired degree of granularity in the feature maps. The size of the output feature maps depends on the quantity of input data, the filter sizes, the padding, and the stride.After the convolutional layer, one or more activation functions are used to offer the network non-linearity and the capacity to learn increasingly complex representations of the input data. The activation function of CNNs for the model frequently employ ReLU, sigmoid, and tanh. ReLU is frequently employed in CNN's hidden layers to add non-linearity and aid in the network's learning of complicated features. ReLU is computationally efficient and successful in avoiding the vanishing gradient problem since it sets negative values to zero and maintains constant positive values. ReLU is useful for computer vision tasks such as image recognition.

When a job requires binary classification, the output layer of a CNN frequently employs the sigmoid activation function. It reduces the input to a probability, which can be understood as a number between 0 and 1. When the model must offer a probability estimate for a binary choice, such as deciding if an image contains a particular object or not, sigmoid is helpful. Tanh is a symmetric activation function that can handle both negative and positive numbers. Tanh condenses the input within the range of -1 to 1. When the output range needs to be symmetric, tanh is frequently utilized in hidden layers of a CNN. This enables the network to recognize more intricate patterns and correlations in the input. Multi-class classification issues often require the use of the softmax activation function. A probability distribution over various classes is produced from an input vector of real numbers. Softmax can be used to calculate the likelihood that an input belongs to each class in a multi-class classification problem because it assures that the total of the output probabilities equals 1.

The pooling layer receives the feature map produced by the convolutional layer and attempts to reduce the spatial size of the feature maps while maintaining the most important data. To do this, the feature maps are divided into non-overlapping sub-regions, and each sub-region output value is pooled to form a single output value. Dropout regularization is a method for avoiding overfitting in neural networks. It entails randomly removing some neurons from a layer during training (i.e., setting them to zero). The output of the layer is given a dropout mask during each training iteration, which randomly sets some of the values to zero. A neuron's chance of dying is normally set between 0.2 and 0.5. According to the gradients of the loss function with respect to the parameters, the optimization algorithm modifies the weights and biases.

The output from the convolutional and pooling layers is flattened into a single vector by the flattening layer. This is required because the input to the Fully Connected layer must be a one-dimensional vector whereas the output from the convolutional layers is often a three-dimensional tensor. In essence, the Flatten layer transforms the tensor into a vector so that it may be transferred to the Fully Connected layer. The extracted characteristics from the preceding layers are processed by the Fully Connected layer, which also creates the output. The Fully Connected layer may understand intricate correlations between the features since every neuron in the lower layer is connected to every neuron in the upper layer. Typically, a softmax layer receives the output from the fully connected layer and provides a probability distribution over the classes in a classification problem. The last stages of a CNN are provided by the Flatten layer and Fully Connected layer working together to process the extracted features and provide the output.



Source : https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/

3 Methodology

Data for this study came from the Kaggle Bird Call Competition, which was originally from Xeno-Canto, a global database of nature sounds. 264 different species of bird calls are represented in the dataset. 12 bird species are chosen for investigation from the data. The project's objective is to use a convolutional neural network (CNN) to determine the bird species from their call. The audio signals are transformed into spectrograms for each 2-second frame in order to apply Convolutional Neural Networks (CNNs) for audio classification. This produces a 343(time) x 256(frequency) image of the bird call. The spectrograms are fed into the model as input, and the response variables are their labels. The data is divided into training and test sets at a ratio of 7:3. To comprehend the various CNN model architectures and their effectiveness, binary classification of numerous bird species is carried out in this study.

3.1 Binary Classification

Based on their calls, the American crow and White-crowned sparrow are predicted by the model. The model receives the pre-processed data for the two classes as input. The CNN model's architecture consists of a number of consecutive layers, each with a convolutional layer's typical kernel size of 3x3 and a pool size of 2x2 for the pooling layer.A convolutional 2D layer with 32 filters and a 3x3 kernel size makes up the top layer. ReLU is the activation function utilized in this layer. The number of frequency bins in the spectrogram is 343; the number of time steps is 256; and the input shape of the layer is (343, 256, 1), where 1 denotes a single color channel. A max pooling layer with a 2x2 pool size is always added after the convolutional layer. The feature maps produced by the convolutional layer are reduced in dimension using this

layer. the subsequent convolution layer, which has filters of 64 and 128 with a middle layer of pooling. The output from the preceding layer is converted into a 1D array in the following flatten layer, which then sends the output to the dense layers with 128 units and 32 units in turn with a ReLU activation. Finally, a dense layer with a single unit and sigmoid activation is the output layer. A binary outcome is predicted using the sigmoid function. The "adam" optimizer, the "binary_crossentropy" loss function, and the "accuracy" metric are used in the model's construction. The model is compiled and trained to fit the data for the epoch of 25 with batch size 15 and finally evaluated the performance on the test set

3.2 Multi-Class Classification

Based on their call, the model can identify 12 different kinds of birds. The model receives the pre-processed data for the two classes as input. The CNN model's architecture consists of a number of consecutive layers, each with a convolutional layer's typical kernel size of 3x3 and a pool size of 2x2 for the pooling layer.A convolutional 2D layer with 32 filters and a 3x3 kernel size makes up the top layer. ReLU is the activation function utilized in this layer. The number of frequency bins in the spectrogram is 343; the number of time steps is 256; and the input shape of the layer is (343, 256, 1), where 1 denotes a single color channel. A max pooling layer with a 2x2 pool size is always added after the convolutional layer. The feature maps produced by the convolutional layer are reduced in dimension using this layer. the subsequent convolution layer, which has filters of (32,64,128,256) with a middle layer of pooling. The output from the preceding layer is converted into a 1D array in the following flatten layer, which then sends the output to the dense layers with (512,256) in turn with a ReLU activation. Finally, a dense layer with a 12 unit and softmax activation is the output. The "adam" optimizer, the "categorical_crossentropy" loss function, and the "accuracy" metric are used in the model's construction.The model is compiled and trained to fit the data for the epoch of 25 with batch size 15 and finally evaluated the performance on the test set

4 Computational Results

4.1 Binary Classification

In order to predict the bird species American crow and white crowned sparrow using binary classification, CNN models with various parameters are trained. As can be seen in Fig. 1, the model with convolution layer (32,64) nodes and dense layer with (32,1) nodes performed well when trained in the batch size of 5 for epoch 25 with the accuracy of 96.7% and the computation time of 26 sec. As a result, the performance of a simpler binary classification model is superior to a complex model.

Model Input Layer	Accuracy	Computation Time
Conv2d(32,64,128) Dense layer(128,32,1)	93.50%	27.6 sec
Conv2d(32,64) Dense layer(32,1)	96.70%	26 sec
Conv2d(32,64,128) Dense layer(64,32,1)	93.50%	27.8 sec
Conv2d(32,64) Dense layer(64,1)	96.70%	30.1 sec
Conv2d(64,128) Dense layer(64,1)	93.50%	63 sec

Fig 1: Binary Classification CNN model performance and computational time

An indicator of a model's performance and its capacity to generalize to fresh, untested data is the test and validation loss. The model can predict the validation set with a similar level of accuracy after learning from the data without overfitting the training set. At epoch 8, the model starts to learn and becomes saturated. Ten epochs are needed to train the model, which reduces computing time even more.

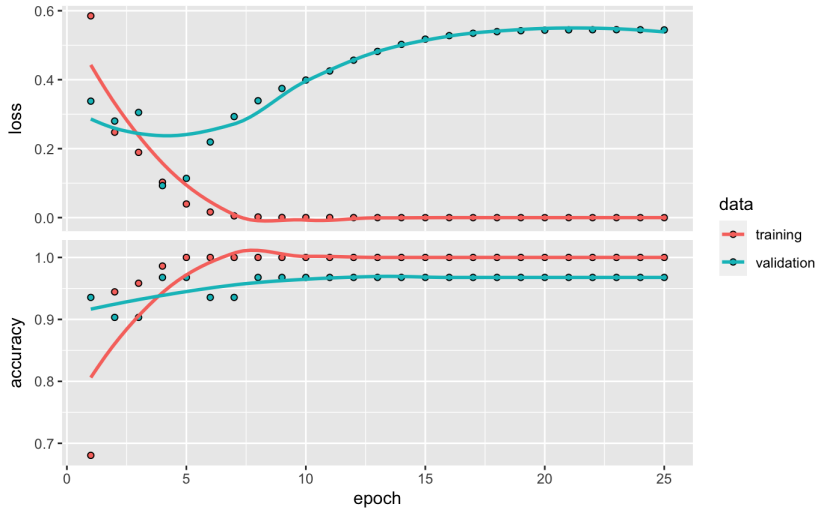


Fig 2: Binary Classification CNN model - function progress during the iteration

The confusion matrix for the CNN model's binary classification is depicted in the figure. The model correctly identified the White-crowned Sparrow 100% of the time while incorrectly identifying the American crow once. For the two species, the model performs admirably. The model can be tested on various species to see how it performs.

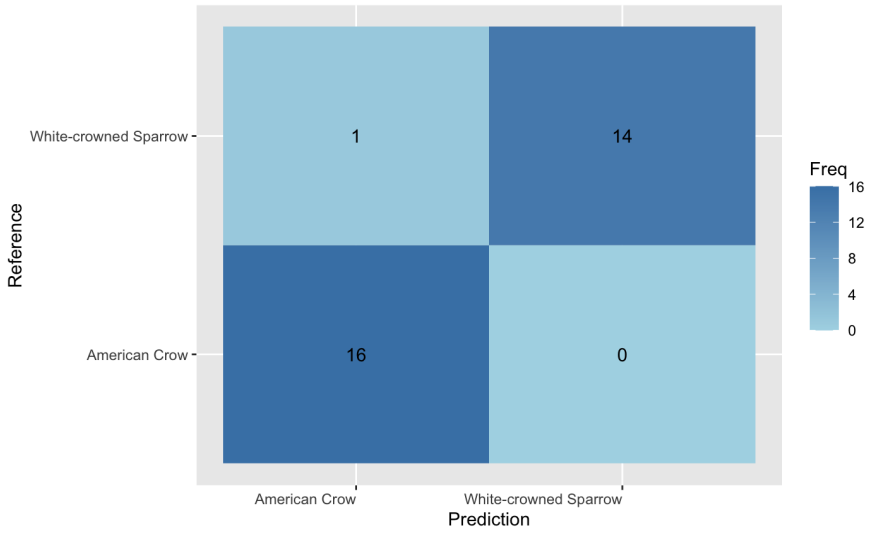


Fig 3: Confusion Matrix - Binary Classification using CNN model

4.2 Multi-Class Classification

In comparison to the binary model, the multi-class model requires more layers to capture the input (call) pattern of 12 different species and will take longer to compute. As a result, the model is adjusted for various parameters, as shown in Fig. 3, and accuracy and computing speed are traded off. As the model tries to capture the features and learn from the input, the computational time grows with the greater number of nodes in the convolution layer and dense layer. The best model has convolutional layers with 32, 64, 128, 256 nodes and dense layers with 512, 256, 12 nodes. It is trained on a test set with 71% accuracy and takes 8 minutes to compute.

Model Input Layer	Accuracy	Computation Time
Conv2d(32,64,128) Dense layer(128,12)	69.50%	318 sec
Conv2d(32,64,128) Dense layer(128,64,12)	68.30%	277sec
Conv2d(32,64,128,256) Dense layer(512,256,12)	71.00%	499 sec
Conv2d(128,256) Dense layer(128,12)	70.00%	13096 sec
Conv2d(64,128) Dense layer(64,12)	71.00%	641 sec

Fig 4:Performance and computational time of classification of CNN model for 12 different species

The model learns from the training data at epoch 20 according to the graph below. Training data exhibit some overfitting, but this can be reduced by incorporating drop out layers.

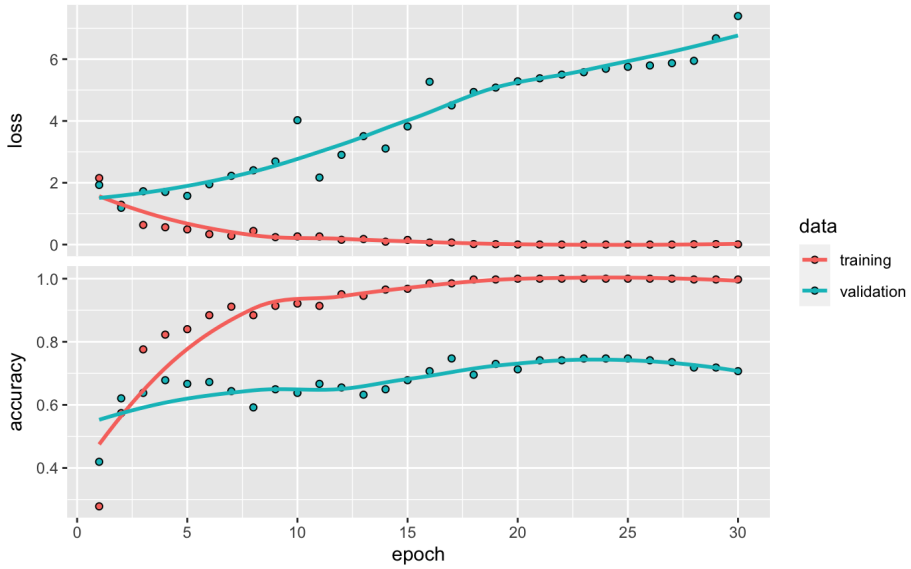


Fig 5: Binary Classification CNN model - function progress during the iteration

The best model's confusion matrix is displayed below. The species Mallard (100%), White-crowned Sparrow (76%), American Crow (71%), Dark-eyed Junco (83%) and Northern Flicker (65%) can all be classified using the model. The spectrogram image's tone and frequency patterns can be seen by the model, which has correctly categorized the data. While the Stellars Jay (40%) and Chickadee (50%) species have shown the model to perform the worst.The model is accurate and does not favor any particular species.

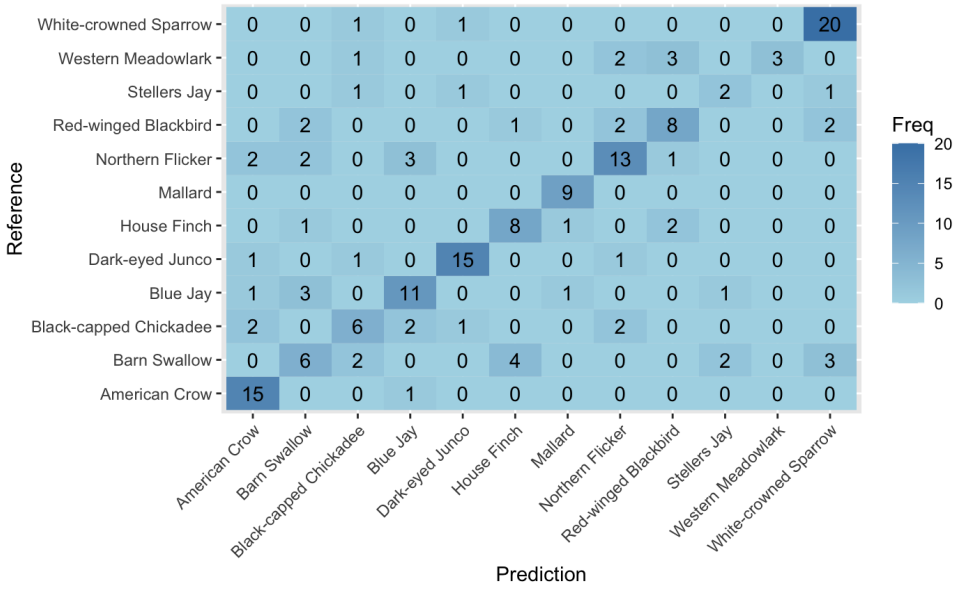


Fig 6:Confusion Matrix - Multi-class classification

5 Discussion

The computational results demonstrate that for the binary classification model, the neural network with a basic topology performs better with a 96.7% accuracy rate and a 26-second computation time. Thus it is always easy to classify two different seasonal birds through CNN Binary classification. Although appropriately built, a complex structure for multi-class models requires a lot of processing (8 mins) and performs fairly with 71% accuracy. It was mistakenly identified as a Western Meadowlark twice and a Red-winged Blackbird twice out of 20 Northern Flickers. There is a possibility that both birds may be present at the same moment, and the model will be able to tell which bird is making the loudest sound. It is difficult to comprehend the reason for the multi-class model's performance. As the sound of the birds can be distorted, adding more data to the input image and through data augmentation, stretching the spectrogram along the time and frequency axes can improve the performance of multi-class models. Additionally,recording an audio with numerous birds chirping simultaneously also affects the model accuracy. This model can be used to track bird migration counts, identify variables influencing bird migration, and conserve threatened seasonal bird species.

6 Conclusion

The goal of this study was to use the Kaggle Bird Call Competition to train a convolutional neural network (CNN) to identify different bird species based solely on their calls. To distinguish the different bird species based on the pitch and tone of their calls, CNN uses audio recordings that have had less transmission loss to create spectrogram images. Different consecutive layers are used to construct the CNN model in order to process and learn from the incoming data. The accuracy of the Binary model was 96%, and the computation took 26 seconds. The accuracy of the multi-class model was 71%, and its computation took 499 seconds. In the future, the input files can be enhanced with data augmentation to improve the model's performance and to experiment with other model parameters, however as we add more layers, the calculation time will grow.

References:

MayankMishra. Convolutional Neural Networks, Explained. Published in Towards Data science. <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>

James, G., Witten, D., Hastie, T., & Tibshirani, R. (n.d.).An Introduction to Statistical Learning. Retrieved April 26, 2023, from https://www.statlearning.com/