SEATTLE UNIVERSITY
1891

Using Neural Network Model to Classify Seattle Bird Species Based on Audio Spectrograms

Report on Machine Learning II

By Erdenetuya Namsrai

2025

Table of Contents

List of Figures

List of Tables

Abstract

In today's era of natural imbalance, protecting the natural balance is the most important issue. In particular, the number of species of birds is decreasing day by day, and to prevent this, we need to accurately count and classify the number of birds and create an ecologically balanced environment for them. To do this, we used the deep learning convolutional neural network method to classify Seattle’s 12 bird species. The dataset consists of spectrograms of 12 types of birds, and binary models and multi-class models were developed. In order to improve the experimental results and prevent overfitting, models with hyperparameters were trained. As a result, the multi-class model improved the accuracy of the training and testing sets compared to the binary classification model, demonstrating the robustness and general ability of the models. In addition, the models were tested on audio recordings to identify the dominant bird species in each recording. The results indicate that machine learning techniques, such as Convolutional Neural Network (CNN), are effective in identifying bird species. The results of this research can be widely used in the fields of ecology and biology to maintain the balance of nature, including preventing the extinction of bird species.

Introduction

Bird species identification is very important in the field of ecology and biology, and by correctly identifying bird species, it is of great importance to maintain the balance of nature, prevent bird extinction, and monitor them. The advancement of information technology, using machine learning methods, has made it possible to easily identify bird species. This research work used the Convolutional Neural Network model, which is used for machine learning image and audio processing, and it is an effective method for classifying bird species from their voices. The research data was taken from the Xeno-Canto bird song archive, which is a site that aims to share recordings of the voices of many wildlife species collected from all over the world. The frequency spectrum over time can be seen in a spectrogram, which is a representation of an audio signal, and it is particularly useful for analyzing bird calls. By converting audio recordings into spectrograms, it is possible to obtain unique acoustic patterns that characterize different bird species. To do this, Convolutional Neural Networks (CNNs) were used, which were applied to these spectrograms to determine the unique characteristics of bird species. CNNs are used in this study to process complicated visual data and automatically recognize bird species based on their vocalizations. The objective of this study is to systematically evaluate CNN models for binary and multi-class bird species classification using spectrograms to determine their accuracy and robustness. The purpose of this study is to develop and evaluate CNN-based models that can categorize binary and multi-class bird species using spectrograms.

Theoretical Background

Deep learning is a type of machine learning, and its main goal is to create models that can think and make predictions like humans. Deep learning is based on neural networks, which are similar in structure to the human brain's nervous system. A neural network consists of the following types of layers:

1. Input Layer
2. Hidden Layer
3. Output Layer

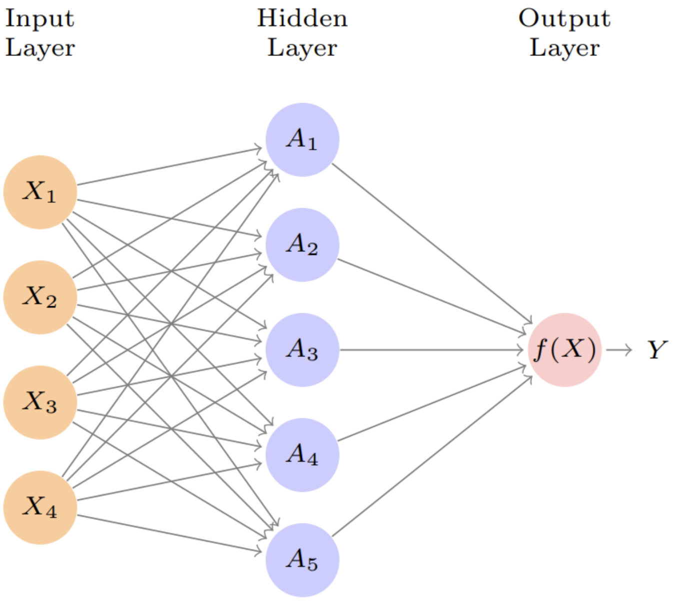


Figure 1. Neural Network Structure

(Source: An introduction to Statistical Learning)

Figure 1. shows neural network structure. In the input layer is the first layer of neural network and it transfers to data from the external environment into network. In other words, input layer receives raw data such as images, text, sound, and numerical data and passes it on to the next layer. In the hidden layer is the most important part and it performs the calculations such as feature recognition, pair formation and trend detection. In other words, neurons respond to input information using weights and activation functions. Also, the hidden layer sends the values to the next layer. The last layer is the output layer, and it outputs the response from the network using a function such as “softmax”, “sigmoid” to produce a probability value.

There are many types of neural networks. The most common types of neural networks are the following, each with its own purpose. These include:

1. Recurrent Neural Network (RNN)
2. Long Short-Term Memory (LSTM)
3. Convolutional Neural Network (CNN)

The Recurrent Neural Network (RNN) can work on the time series data, and the main feature is that it can remember information from the previous steps in a sequence and use that to influence the current output.

The Long Short-Term Memory (LSTM) is an improved version of Recurrent Neural Network, which is a deep network capable of maintaining long term dependencies.

In this research, we used Convolutional Neural Network (CNN), which is used to detect features in data such as images, sounds, images, and videos, and to classify and predict them. With the help of this convolutional neural network, we used it to identify or classify bird species based on bird songs based on sound images. The main goal of CNN is to determine the features and understand and classify patterns and structures in the input data.

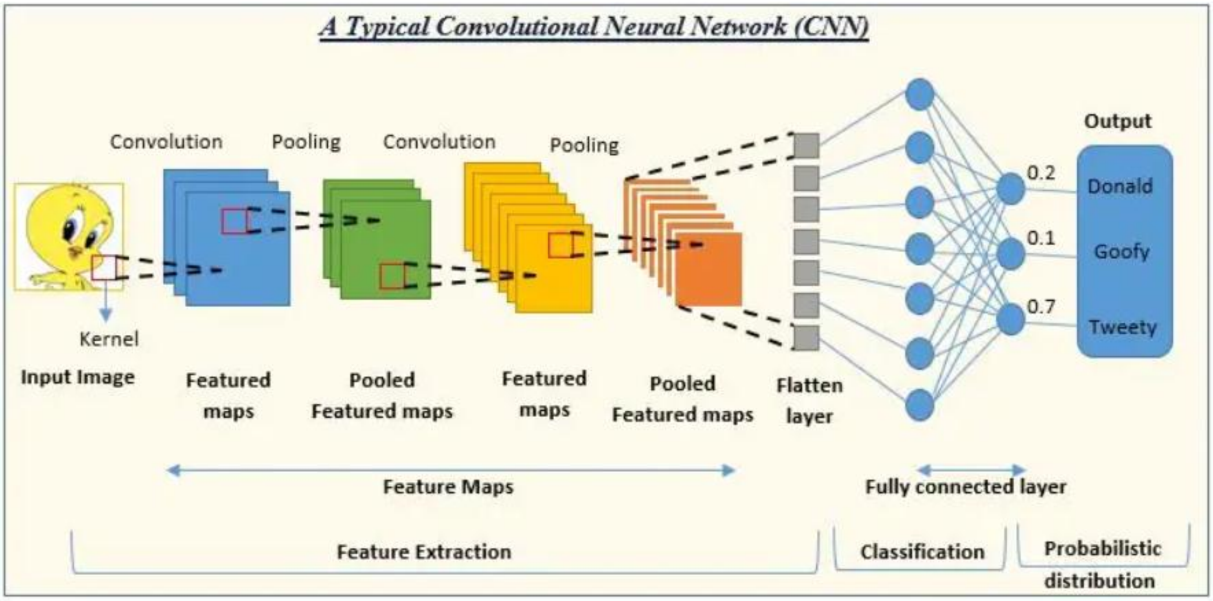


Figure 2. Convolutional Neural Network Structure

(Source: https://glasswing.vc/blog/ai-atlas-16-convolutional-neural-networks-cnns/)

Figure 2. shows the structure of a Convolutional Neural Network, which consists of the following parts.

№	Layers	Description
1.	Input layer	Receive the raw data
2.	Convolutional Layer	Filters or kernels are used to convolve an image. The kernels detect local features in an image.
3.	ReLU	After convolution, an activation function is applied to introduce non-linearity, which helps the network learn complex patterns.
4.	Pooling layer	Reduce the spatial size of the feature maps and it helps to reduce computation, prevent overfitting, and keep the most important information.
5.	Flatten layer	Convert the 2D feature maps into a 1D vector.
6.	Fully connected layer	Every neuron is connected, and it combines all features to make predictions.
7.	Output layer	Make the final prediction

Table 1. Convolutional Neural Network

A Convolutional Neural Network has parameters that perform specific functions to enable accurate classification.

№	Category	Parameters	Description
1.	CNN Model architecture	Number of layers	How deep the network is
		Neurons per layer	Controls model complexity
		Type of layers	Dense, Convolutional, Recurrent
2.	Activation Function	ReLU, Sigmoid, Softmax, and others	Converts to data entered into the network into a specific form and decide whether the neuron will be activated. In other words, activation function converts the data into a nonlinear form, making the network more accurate, powerful, and capable of learning
3.	Optimizer	Adam, RMSprop, and others	Updates weights during training
4.	Loss Function	Cross-entropy, MSE, and others	Measures model error
5.	Learning rate	0.1, 0.01, 0.001, and others	Controls how fast weights update

6.	Regularization	Dropout rate, L1/L2 penalty	Helps prevent overfitting
7.	Batch size	16, 32, 64, and others	Number of samples per training step
8.	Epochs	10, 50, and 100, and others	How many times the model sees the data
9.	Weight initialization	He, Xavier, Random	Affects early convergence and training stability

Table 2. Convolutional Neural Network's Parameter

When developing and training neural network models, we first define the problem and select the data. Next, we build a model and train it. After training, we evaluate the model, and if performance needs improvement, we tune the model by adjusting hyperparameters and experimenting with different architectures. In addition, the following model evaluation metrics were used to measure classification performance.

1. A Confusion Matrix is a table that describes the performance of a classification model by displaying the true positives, false positives, true negatives, and false negatives.
2. Accuracy is the proportion of correctly classified cases. In other words, it is the main metric for evaluating classification.
3. Precision is the proportion of true positive predictions among all positive predictions.
4. Recall is the proportion of true positive predictions among all actual positives.
5. F1-score is the harmonic mean of precision and recall.

Methodology

This study aims to classify bird species using Convolutional Neural Networks applied to spectrograms of bird calls. The methodology includes data preparation, model development of binary model, multi-class model, training, evaluation, and comparison of different neural network architectures.

1. Data preparation

The dataset consists of spectrograms for 12 bird species stored in an HDF5 file. Each spectrogram represents an audio recording of bird calls converted to the frequency domain using the Short-Time Fourier Transform.

2. Test/Train Split The

Dataset is divided into training and testing sets using a 70/30 split, meaning 70% of the data is used for training the model, and 30% is reserved for evaluating its performance.

3. Variables and Predictors

The input variables for the CNN are the pixel values of the spectrograms. Each spectrogram serves as a feature matrix representing the frequency content of the bird calls over time. The output variable is the bird species label, encoded using label encoding for binary classification and one-hot encoding for multi-class classification. The model is designed to predict the probability distribution across the following Seattle's 12 bird species:

Figure 3. 12 of Seattle's bird species
(Source: https://seattleu.instructure.com/)

- American Crow (amecro),
- American Robin (amerob),
- Bewick's Wren (bewwre),
- Black-capped Chickadee (bkcchi)
- Dark-eyed Junco (daejun),
- House Finch (houfin),
- House Sparrow (houspa),
- Northern Flicker (norfli),
- Red-winged Blackbird (rewbla),
- Song Sparrow (sonspa),
- Spotted Towhee (spotow),
- White-crowned Sparrow (whcspa)

4. Model Development

Three CNN architectures were developed and evaluated in this study, each for the binary classification model and the multi-classification model. The binary classification model is designed to distinguish between two bird species, American Crow (amecro) and American Robin (amerob). This binary CNN model includes convolutional layers with ReLU activation, max-pooling layers, and fully connected layers with a "sigmoid" activation function in the output layer to predict binary outcomes. It also includes a configurable optimizer and an early stopping callback to prevent overfiring and improve generalization performance of the binary CNN model. In the multi-class CNN model extends the architecture to classify 12 bird species using a "softmax" activation function in the output layer to predict the probability distribution across multiple classes.

4.1 Binary Classification

In the binary classification task, spectrograms for two bird species, American Crow (amecro) and American Robin (amerob) were used to train a CNN model. The data pipeline included the following steps such as converting data into an appropriate format, value normalization, class balancing, and data augmentation. These steps helped address class imbalance, reduce overfitting, and improve the performance and generalization of Binary CNN models.

Three different binary CNN models are developed and trained with:

4.1.1 A basic binary CNN model1 consisting of convolutional layers with ReLU activation, max-pooling layers, and fully connected layers with a sigmoid activation function in the output layer to predict binary outcomes.

4.1.2 The next binary CNN model2 includes dropout and batch normalization layers to prevent overfitting.

4.1.3 Tuned CNN model3 with both dropout, batch normalization, early stopping, and L2 regularization. L2 regularization adds a penalty term to the loss function to discourage large weights, further helping to prevent overfitting.

4.2 Multi-class Classification

Multi-Class Classification with Limited Samples For the multi-class classification task, spectrograms for all 12 bird species are loaded, but only the first 30 samples for each species are used initially. This approach ensures a balanced dataset and allows for quicker experimentation and model tuning. Using a smaller, balanced dataset helps in identifying the best model architecture and hyperparameters without requiring extensive computational resources. Three different multi-class CNN models are developed and trained using these limited samples:

4.2.1 A basic multi-class CNN model is designed to classify 12 bird species using convolutional layers with ReLU activation, max-pooling layers, and fully connected layers with a softmax activation function in the output layer.

4.2.2 The second multi-class CNN model that includes dropout layers and to prevent overfitting.

4.2.3 Tuned CNN model with both dropout, batch normalization, early stopping, and L2 regularization to improve generalization.

4.3 Model Training

The models are compiled and trained using the rmsprop and adam optimizers, which adjust the learning rate dynamically to ensure efficient convergence. The binary classification models are trained for 10, 30, and 50 epochs, while the multi-class models are trained for 20, 50, and 100 epochs. These fixed epoch counts allow for controlled comparisons of the model performance at different stages of training. Although a model trained for 50 epochs passes through earlier epochs, training separate models ensures consistent evaluation, especially if early stopping doesn't happen. Additionally, early stopping is employed to halt training if the validation loss does not improve for five consecutive epochs, preventing overfitting and ensuring optimal performance.

4.4 Model Evaluation and Comparison

Both binary and multi-class CNN models are evaluated using several metrics, including accuracy, precision, recall, F1 score, and confusion matrix. Accuracy measures the proportion of correctly classified instances out of the total instances, while precision, recall, and F1 score provide insights into the model's performance, especially in 12 imbalanced datasets. The confusion matrix visualizes the performance across different classes, highlighting true positives, false positives, true negatives, and false negatives.

4.5 Evaluation of Test Set

After training, the models are evaluated on the test set to assess their generalization capabilities. The test accuracy, precision, recall, and F1 score are calculated and compared across different models. The final model's performance on the test set indicates its effectiveness in classifying unseen bird species spectrograms.

5. Testing on New Audio Clips

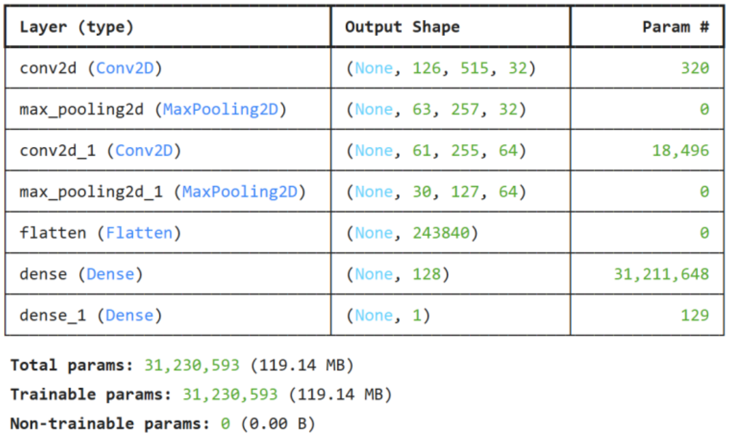
Three audio clips were tested to evaluate the best multi-class CNN model on unsampled data. Each clip is subsampled to 22050 Hz, and loud segments are extracted into 2-second windows. The model predicts the bird species for each spectrogram and combines the predictions to identify the most likely bird species or names for each clip.

Computational Results

The computational results for the binary and multi-class classification tasks are presented below. The performance of the models is evaluated using accuracy, precision, recall, F1 score, and confusion matrix. These metrics provide a comprehensive view of how well the models classify bird species from spectrograms.

1. Binary CNN classification models results

1.1 Basic Binary CNN Model_1



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 515, 32)	320
max_pooling2d (MaxPooling2D)	(None, 63, 257, 32)	0
conv2d_1 (Conv2D)	(None, 61, 255, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 127, 64)	0
flatten (Flatten)	(None, 243840)	0
dense (Dense)	(None, 128)	31,211,648
dense_1 (Dense)	(None, 1)	129

```
Total params: 31,230,593 (119.14 MB)
Trainable params: 31,230,593 (119.14 MB)
Non-trainable params: 0 (0.00 B)
```

Figure 4. Basic Binary CNN Model_1

Basic binary CNN model_1 features two Conv2D layers each followed by a max-pooling2D layer, flatten layer, and dense layer with 128 neurons. That model has 31230593 trainable parameters, and it indicates its complexity and capacity for learning.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
84%	98.33%	84.72%	0.63

Table 3. Binary CNN model's performance

Table 1 shows the performance of Binary CNN model. Accuracy rate is 84% and it indicates the model correctly classified 84% of the test samples. Training accuracy is 98.33% and it indicates the model correctly classified all training samples. However, the significant gap between that training and test accuracy suggests that the model may be overfitting to the training data.

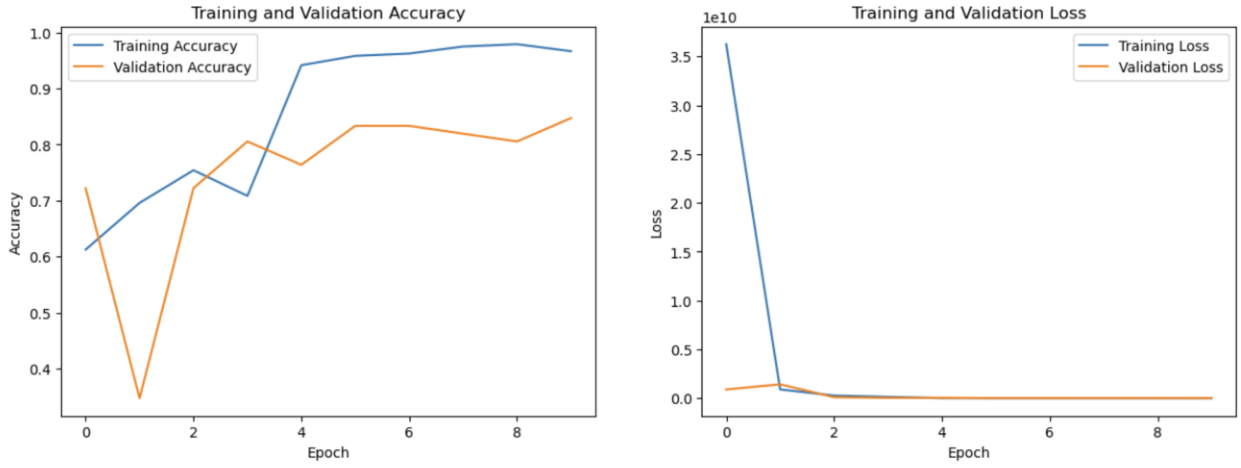


Figure 5. Basic Binary CNN Model_1's Performance of Training and Val accuracy vs Training and Val Loss

These plots show strong learning on the training data, reaching near-perfect accuracy with very low training loss. However, the validation accuracy is noticeably lower and fluctuates across epochs, while validation loss remains steady. This suggests that the model may be overfitting performing well on the training set but not generalizing as well to new data.

Actual/Predicted	**American Crow**	**American Robin**
American Crow	9 (TN)	11 (FP)
American Robin	0 (FN)	52 (TP)

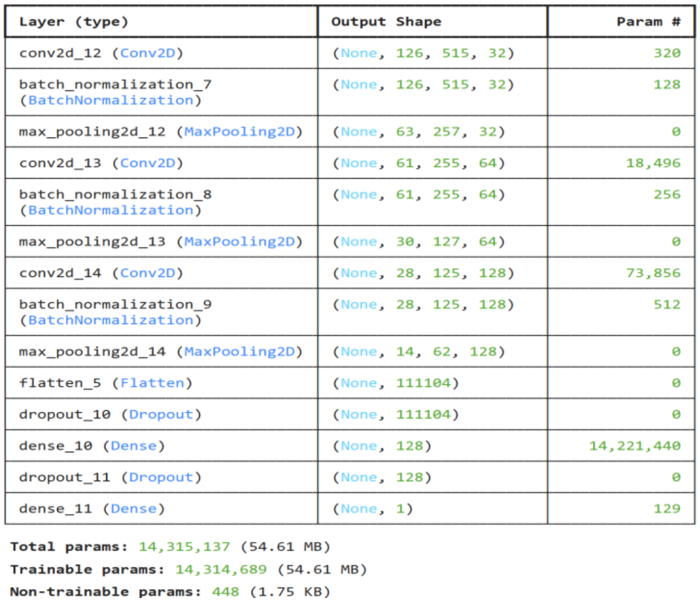
Table 4. Basic Binary CNN model's confusion matrix

Model	***Precision***		***Recall***		***F1-Score***		***Accuracy***
	American Crow	***American Robin***	***American Crow***	***American Robin***	***American Crow***	***American Robin***	
Basic Binary CNN model_1	*100%*	*83%*	*45%*	*100%*	*62%*	*90%*	***78.4%***

Table 5. Basic Binary CNN model's performance evaluation

The results show that the model is very good at detecting American Robin, with a recall of 100%, while it detects American Crow with 45% recall, meaning many American Crow cases are misclassified. However, the model still achieves a high overall accuracy of 85% and has balanced weighted performance metrics.

1.2 Binary CNN Model_2



```
Layer (type)                        Output Shape                   Param #
conv2d_12 (Conv2D)                  (None, 126, 515, 32)               320
batch_normalization_7               (None, 126, 515, 32)               128
(BatchNormalization)
max_pooling2d_12 (MaxPooling2D)     (None, 63, 257, 32)                  0
conv2d_13 (Conv2D)                  (None, 61, 255, 64)             18,496
batch_normalization_8               (None, 61, 255, 64)                256
(BatchNormalization)
max_pooling2d_13 (MaxPooling2D)     (None, 30, 127, 64)                  0
conv2d_14 (Conv2D)                  (None, 28, 125, 128)            73,856
batch_normalization_9               (None, 28, 125, 128)               512
(BatchNormalization)
max_pooling2d_14 (MaxPooling2D)     (None, 14, 62, 128)                  0
flatten_5 (Flatten)                 (None, 111104)                       0
dropout_10 (Dropout)                (None, 111104)                       0
dense_10 (Dense)                    (None, 128)                 14,221,440
dropout_11 (Dropout)                (None, 128)                          0
dense_11 (Dense)                    (None, 1)                          129

Total params: 14,315,137 (54.61 MB)
Trainable params: 14,314,689 (54.61 MB)
Non-trainable params: 448 (1.75 KB)
```

Figure 6. Second Binary CNN Model_2

The second binary CNN model_2 features three Conv2D layers, each followed by a max-pooling2D layer, flatten layer, dense layer, batch normalization, and dropout layer to reduce overfitting and improve performance. That model has 14314689 trainable parameters, and it indicates its complexity and capacity for learning.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
82%	93.75%	82%	5.6

Table 6. Second Binary CNN model's performance

Table 4 shows the performance of the second CNN binary model achieved a training accuracy of 93.75% and a good test accuracy of 82%, indicating effective learning and good generalization. Although there is a small overfitting gap, the use of dropout and Adam optimizer helped maintain solid test performance.

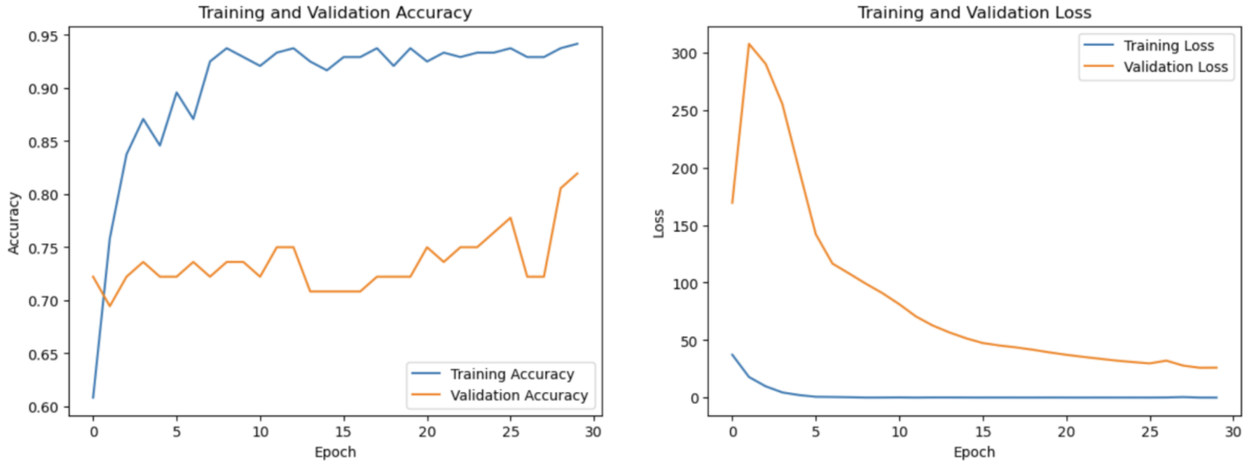


Figure 7. Second Binary CNN Model_2's Performance of Training and Val accuracy vs Training and Val Loss

The second binary CNN model shows strong learning behavior, achieving nearly 93% accuracy on the training set and stabilizing around 82% on the validation set. The sharp drop and stabilization of both training and validation losses indicate that the model converged effectively. However, the consistent performance gap points to mild overfitting.

Actual/Predicted	**American Crow**	**American Robin**
American Crow	13 (TN)	7 (FP)
American Robin	6 (FN)	46 (TP)

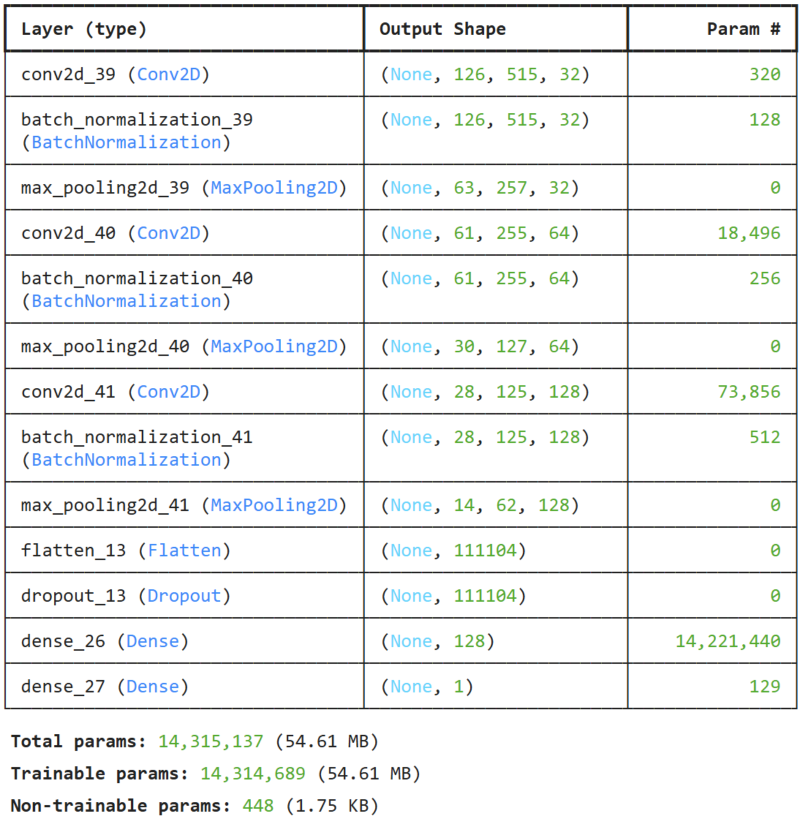
Table 7. Second Binary CNN model's confusion matrix

Model	***Precision***		***Recall***		***F1-Score***		***Accuracy***
	American Crow	***American Robin***	***American Crow***	***American Robin***	***American Crow***	***American Robin***	
Second Binary CNN model_2	*68%*	*87%*	*65%*	*88%*	*67%*	*88%*	***82%***

Table 8. Second Binary CNN model's performance evaluation

The Second CNN model achieved test accuracy of 82% and performed in identifying the class American Robin, with a recall of 88% and an F1-score of 88%. While performance on the class American Crow improved, the model still showed weaker generalization to this class, with a recall of 65% and an F1-score of 67%.

1.3 Tuned Binary CNN Model_3



```
Layor (type)                        Output Shape               Param #
conv2d_39 (Conv2D)                  (None, 126, 515, 32)           320
batch_normalization_39              (None, 126, 515, 32)           128
(BatchNormalization)
max_pooling2d_39 (MaxPooling2D)     (None, 63, 257, 32)              0
conv2d_40 (Conv2D)                  (None, 61, 255, 64)         18,496
batch_normalization_40              (None, 61, 255, 64)            256
(BatchNormalization)
max_pooling2d_40 (MaxPooling2D)     (None, 30, 127, 64)              0
conv2d_41 (Conv2D)                  (None, 28, 125, 128)        73,856
batch_normalization_41              (None, 28, 125, 128)           512
(BatchNormalization)
max_pooling2d_41 (MaxPooling2D)     (None, 14, 62, 128)              0
flatten_13 (Flatten)                (None, 111104)                   0
dropout_13 (Dropout)                (None, 111104)                   0
dense_26 (Dense)                    (None, 128)             14,221,440
dense_27 (Dense)                    (None, 1)                      129

Total params: 14,315,137 (54.61 MB)
Trainable params: 14,314,689 (54.61 MB)
Non-trainable params: 448 (1.75 KB)
```

Figure 8. Tuned Binary CNN Model_3

The tuned binary CNN model_3 features three Conv2D layers, each followed by a max-pooling2D layer, flatten layer, dense layer, batch normalization, early stopping, and dropout layer to reduce overfitting and improve performance. That model has 14315137 trainable parameters, and it indicates its complexity and capacity for learning.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
89.5%	91%	85%	10.23

Table 9. Tuned Binary CNN model's performance

This tuned binary CNN model achieved a training accuracy of 91% and a test accuracy of 85%, demonstrating that the model has effectively learned from the training data and generalizes reasonably well to unseen data. The relatively small gap between training and test accuracy suggests that overfitting is under control and the model maintains good generalization performance.

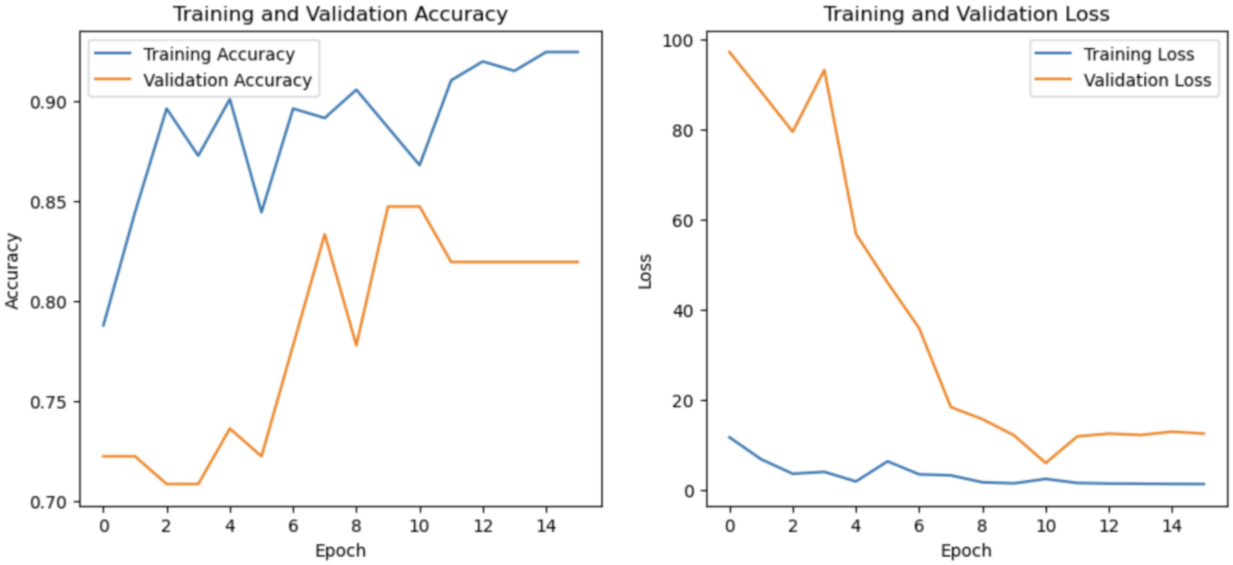


Figure 9. Tuned Binary CNN Model_3's Performance of Training and Val accuracy vs Training and Val Loss

The tuned binary CNN model, enhanced with regularization, balanced training, and early stopping, achieved strong training performance and improved validation accuracy.

Actual/Predicted	**American Crow**	**American Robin**
American Crow	11 (TN)	9 (FP)
American Robin	2 (FN)	50 (TP)

Table 10. Tuned Binary CNN model's confusion matrix

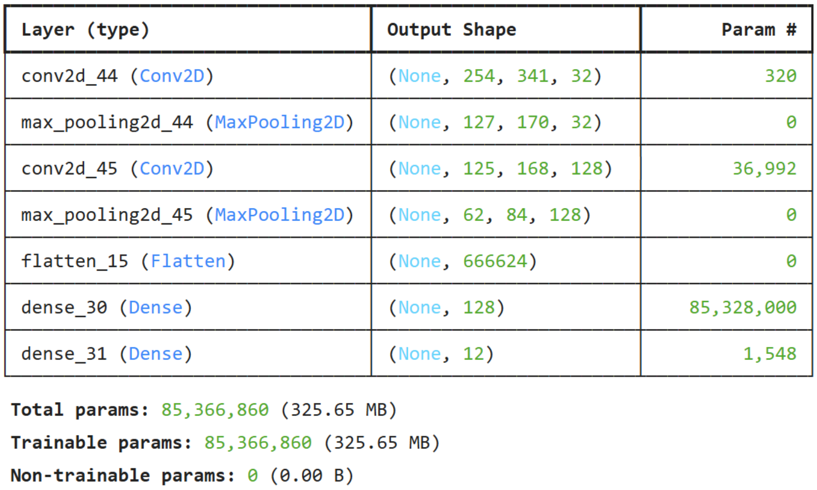
Model	***Precision***		***Recall***		***F1-Score***		***Accuracy***
	American Crow	***American Robin***	***American Crow***	***American Robin***	***American Crow***	***American Robin***	
Tuned Binary CNN model_3	*85%*	*85%*	*55%*	*96%*	*67%*	*90%*	***85%***

Table 11. Tuned Binary CNN model's performance evaluation

The tuned binary CNN model achieved an overall accuracy of 85% and demonstrated balanced precision across both classes. It performed strongly in identifying the majority class American Robin, with a recall of 96% and an F1-score of 90%. However, the model struggled to detect the minority class American Crow, with a lower recall of 55% and F1-score of 67%.

2. Multi-class CNN classification models results

2.1 Basic Multi-class CNN Model_1



```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                      ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ conv2d_44 (Conv2D)                │ (None, 254, 341, 32)   │           320 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_44 (MaxPooling2D)   │ (None, 127, 170, 32)   │             0 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_45 (Conv2D)                │ (None, 125, 168, 128)  │        36,992 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_45 (MaxPooling2D)   │ (None, 62, 84, 128)    │             0 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten_15 (Flatten)              │ (None, 666624)         │             0 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_30 (Dense)                  │ (None, 128)            │    85,328,000 │
├───────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_31 (Dense)                  │ (None, 12)             │         1,548 │
└───────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 85,366,860 (325.65 MB)
 Trainable params: 85,366,860 (325.65 MB)
 Non-trainable params: 0 (0.00 B)
```

Figure 10. Basic Multi-class CNN Model_1

Basic Multi-class CNN model_1 includes two Conv2D layers with 32 and 128 filters respectively, each followed by a MaxPooling2D layer for down sampling. A Flatten layer converts the 3D feature maps to a 1D vector, which is then processed by a dense layer with 128 neurons. The final Dense layer has 12 neurons. The model has a total of 85366860 trainable parameters.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
69.13%	99.34%	69.14%	10.58

Table 12. Basic Multiclass CNN model's performance

Basic multi-class CNN model reached 99.34% accuracy on training data and 69.14% on test data, which is just slightly better than random guessing for 12 classes. This means the model did not learn well and likely needs better training data, a stronger model, or improved input features to perform better.

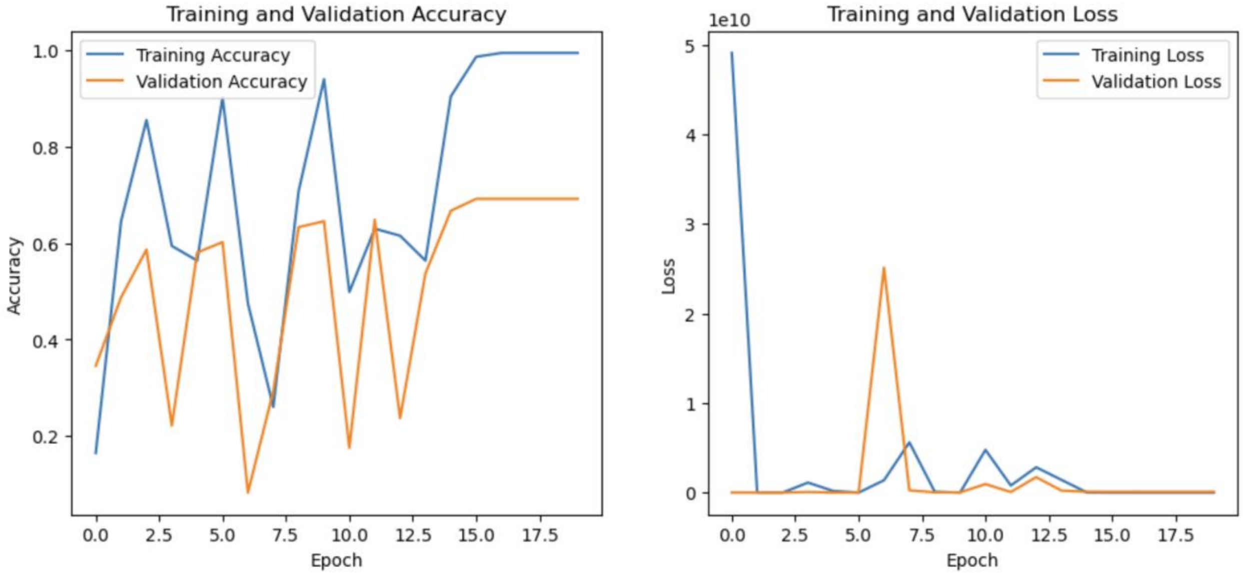


Figure 11. Basic multi-class CNN Model_1's Performance of Training and Val accuracy vs Training and Val Loss

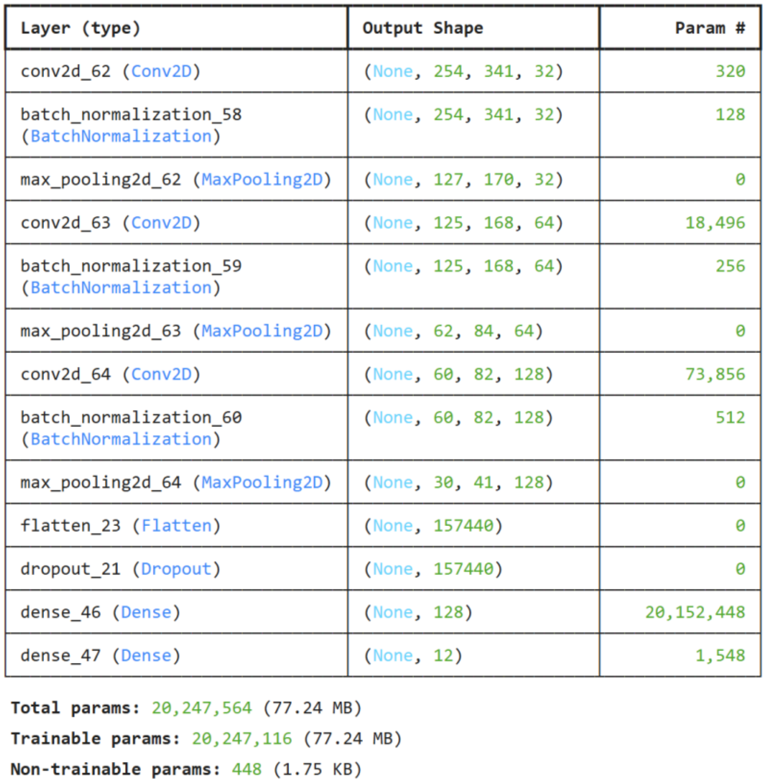
Although the plots show strong performance on the training data, they display unstable and fluctuating results on the validation set, which may indicate overfitting and limited generalization to unseen data.

Basic Multi-class Model_1	**Precision**	**Recall**	**F1-Score**	**Accuracy**
American Crow	95%	67%	78%	**69%**
American Robin	79%	85%	82%	
Bewick's Wren	48%	52%	50%	
Black-capped Chickadee	77%	85%	81%	
Dark-eyed Junco	56%	67%	61%	
House Finch	75%	67%	71%	
House Sparrow	45%	67%	54%	
Northern Flicker	85%	63%	72%	
Red-winged Blackbird	96%	96%	96%	
Song Sparrow	69%	41%	51%	
Spotted Towhee	71%	81%	76%	
White-crowned Sparrow	59%	59%	59%	

Table 13. Basic Multi-class CNN model's performance evaluation

In the results, the basic multi-class CNN model achieved an overall accuracy of 69% across 12 bird species. These birds can be detected by the model, as indicated by amecro, amerob, bkcchi, houfin, norfli, rewbla, and spotow.

2.2 Multi-class CNN Model_2



```
Layer (type)                        Output Shape                    Param #
conv2d_62 (Conv2D)                  (None, 254, 341, 32)                320
batch_normalization_58              (None, 254, 341, 32)                128
(BatchNormalization)
max_pooling2d_62 (MaxPooling2D)     (None, 127, 170, 32)                  0
conv2d_63 (Conv2D)                  (None, 125, 168, 64)             18,496
batch_normalization_59              (None, 125, 168, 64)                256
(BatchNormalization)
max_pooling2d_63 (MaxPooling2D)     (None, 62, 84, 64)                    0
conv2d_64 (Conv2D)                  (None, 60, 82, 128)              73,856
batch_normalization_60              (None, 60, 82, 128)                 512
(BatchNormalization)
max_pooling2d_64 (MaxPooling2D)     (None, 30, 41, 128)                   0
flatten_23 (Flatten)                (None, 157440)                        0
dropout_21 (Dropout)                (None, 157440)                        0
dense_46 (Dense)                    (None, 128)                  20,152,448
dense_47 (Dense)                    (None, 12)                        1,548

Total params: 20,247,564 (77.24 MB)
Trainable params: 20,247,116 (77.24 MB)
Non-trainable params: 448 (1.75 KB)
```

Figure 12. Next Multi-class CNN Model_2

The second Multi-class CNN model_2 includes three Conv2D layers with 32, 64 and 128 filters respectively, each followed by a MaxPooling2D layer for down sampling. A Flatten layer converts the 3D feature maps to a 1D vector, which is then processed by a dense layer with 128 neurons. The final Dense layer has 12 neurons. The model has a total of 20247564 trainable parameters.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
72%	95.4%	72%	53.89

Table 14. Second Multiclass CNN model's performance

The second Multi-class CNN model reached 95.4% accuracy on training data and 72% on test data. The test accuracy reveals a performance drop on unseen data, which suggests overfitting. While the model generalizes moderately well than the previous model.

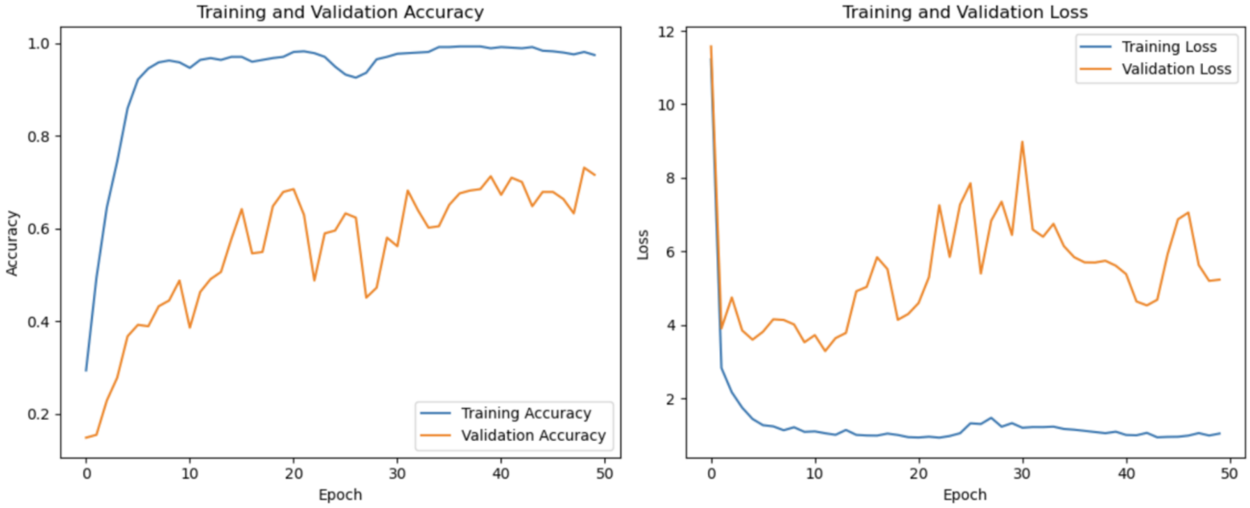


Figure 13. Second multi-class CNN Model_1's Performance of Training and Val accuracy vs Training and Val Loss

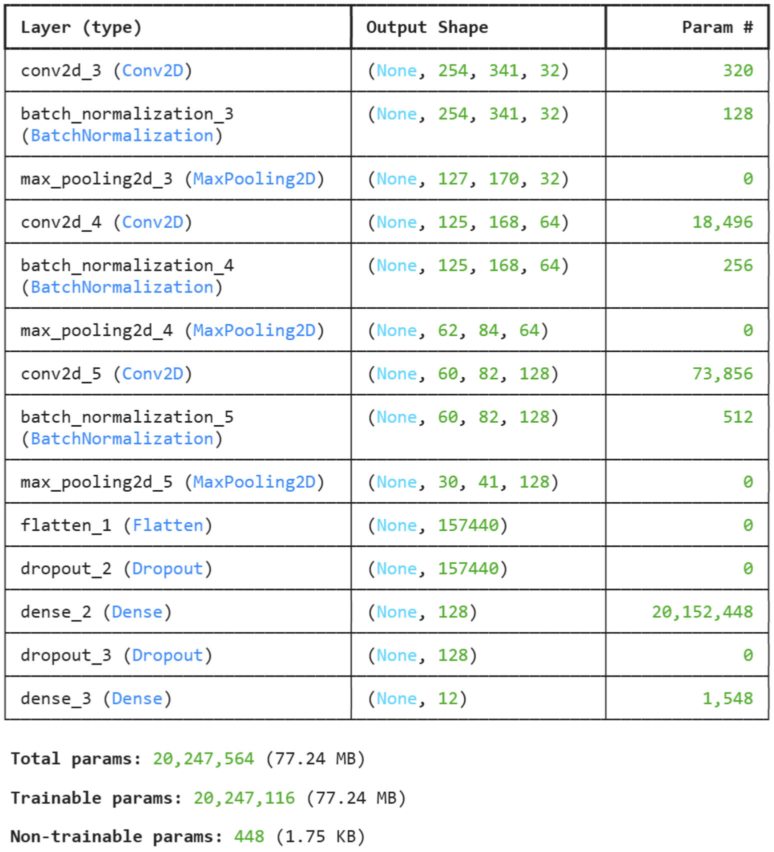
Although this plot shows very high training accuracy, validation accuracy improves slowly and fluctuates, resulting in high validation loss. This indicates overfitting. The model learns well from the training data but struggles to generalize to unseen data.

Second Multi-class Model_2	Precision	Recall	F1-Score	Accuracy
American Crow	94%	56%	70%	**72%**
American Robin	71%	93%	81%	
Bewick's Wren	69%	74%	71%	
Black-capped Chickadee	67%	96%	79%	
Dark-eyed Junco	90%	70%	79%	
House Finch	71%	63%	67%	
House Sparrow	75%	67%	71%	
Northern Flicker	100%	56%	71%	
Red-winged Blackbird	61%	81%	70%	
Song Sparrow	67%	67%	67%	
Spotted Towhee	62%	78%	69%	
White-crowned Sparrow	67%	59%	63%	

Table 15. Second Multi-class CNN model's performance evaluation

The multi-class CNN model achieved an overall accuracy of 72% across 12 bird species. These birds can be detected by models, as indicated by amerob, bkcchi, and daejun.

2.3 Tuned Multi-class CNN Model_3



```
Layer (type)                        Output Shape                 Param #
conv2d_3 (Conv2D)                   (None, 254, 341, 32)             320
batch_normalization_3               (None, 254, 341, 32)             128
(BatchNormalization)
max_pooling2d_3 (MaxPooling2D)      (None, 127, 170, 32)               0
conv2d_4 (Conv2D)                   (None, 125, 168, 64)          18,496
batch_normalization_4               (None, 125, 168, 64)             256
(BatchNormalization)
max_pooling2d_4 (MaxPooling2D)      (None, 62, 84, 64)                 0
conv2d_5 (Conv2D)                   (None, 60, 82, 128)           73,856
batch_normalization_5               (None, 60, 82, 128)              512
(BatchNormalization)
max_pooling2d_5 (MaxPooling2D)      (None, 30, 41, 128)                0
flatten_1 (Flatten)                 (None, 157440)                     0
dropout_2 (Dropout)                 (None, 157440)                     0
dense_2 (Dense)                     (None, 128)               20,152,448
dropout_3 (Dropout)                 (None, 128)                        0
dense_3 (Dense)                     (None, 12)                     1,548

Total params: 20,247,564 (77.24 MB)
Trainable params: 20,247,116 (77.24 MB)
Non-trainable params: 448 (1.75 KB)
```

Figure 14. Tuned Multi-class CNN Model_3

The tuned multi-class CNN model_3 features three Conv2D layers, each followed by a max-pooling2D layer, flatten layer, dense layer, batch normalization, early stopping, and dropout layer to reduce overfitting and improve performance. That model has 20248564 trainable parameters, and it indicates its complexity and capacity for learning.

Accuracy	Training Accuracy	Test Accuracy	Training time (min)
73%	93%	73%	54

Table 16. Tuned Multiclass CNN model's performance

The tuned Multi-class CNN model reached 93% accuracy on training data and 73% on test data. This indicates strong learning during training but some degree of overfitting. However, the test accuracy of 73% aligns with the multi-class classification report, showing consistent generalization on unseen data.

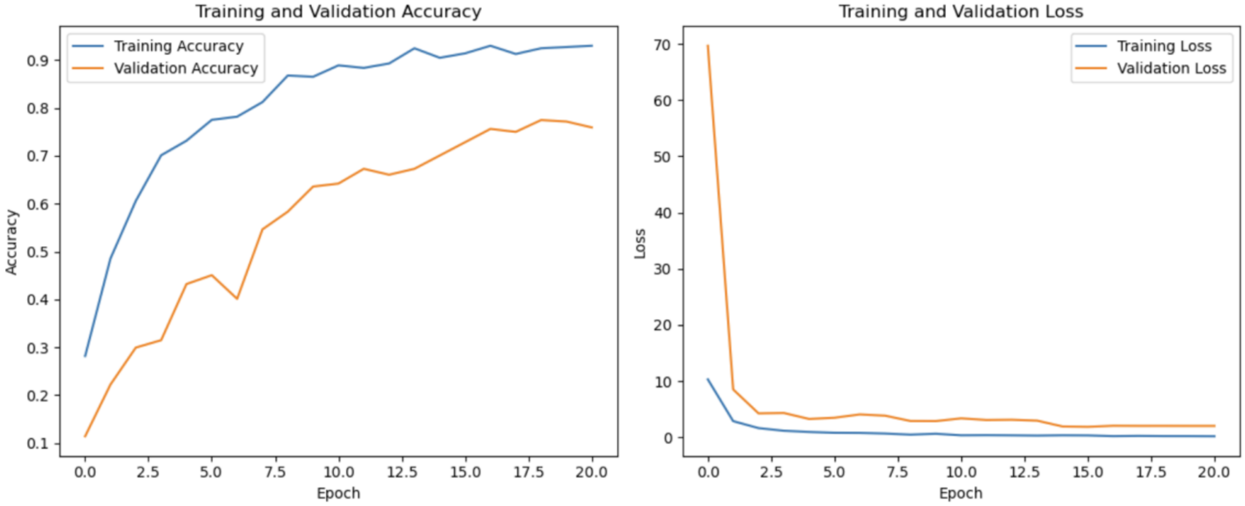


Figure 15. Tuned Multi-class CNN Model_1's Performance of Training and Val accuracy vs Training and Val Loss

The tuned multi-class CNN model demonstrates strong learning ability and good generalization. Although validation accuracy is slightly less than training, the steady upward trend and steady loss indicate a well-trained and reliable CNN for identifying multiple bird species.

Tuned Multi-class Model_3	**Precision**	**Recall**	**F1-Score**	**Accuracy**
American Crow	91%	74%	82%	**73%**
American Robin	78%	78%	78%	
Bewick's Wren	76%	48%	59%	
Black-capped Chickadee	71%	89%	79%	
Dark-eyed Junco	83%	89%	86%	
House Finch	56%	67%	61%	
House Sparrow	63%	63%	63%	
Northern Flicker	76%	70%	73%	
Red-winged Blackbird	71%	89%	79%	
Song Sparrow	76%	59%	67%	
Spotted Towhee	71%	81%	76%	
White-crowned Sparrow	72%	67%	69%	

Table 17. Tuned Multi-class CNN model's performance evaluation

The Tuned multi-class CNN model achieved an overall accuracy of 73% across 12 bird species. The model performed best on species such as daejun and amecro are showing both high precision, F1-score, and recall. Conversely, species like bewwre and houfin were more difficult to classify.

3. External Test Data

Our trained 12 species CNN model and segmenting each external test clip allowed us to identify which bird species were present in every audio file. We used the predictions about the dominant species across segments for each clip to infer if more than one bird was calling. If more than one dominant species appeared, we concluded that the clip contains multiple bird calls. Also, if predictions were consistently from a single species, we determined that the clip likely contains only one bird.

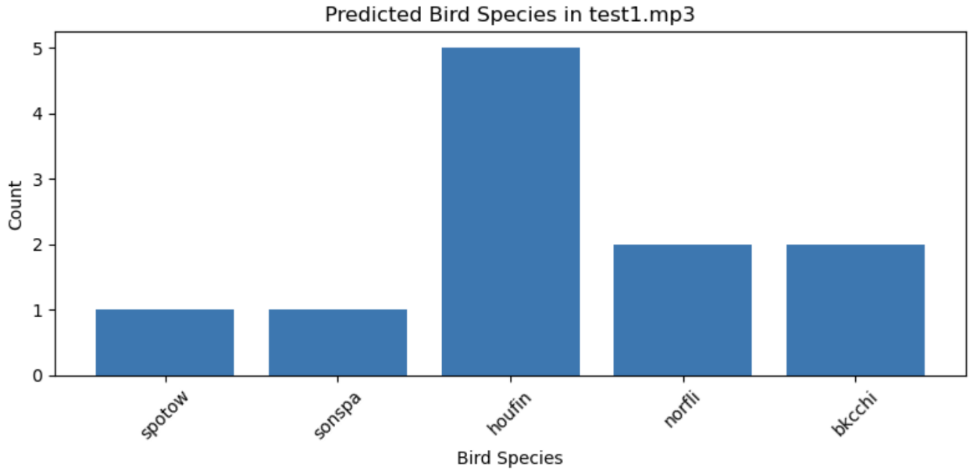


Figure 16. Predicted bird species in Test1.mp3

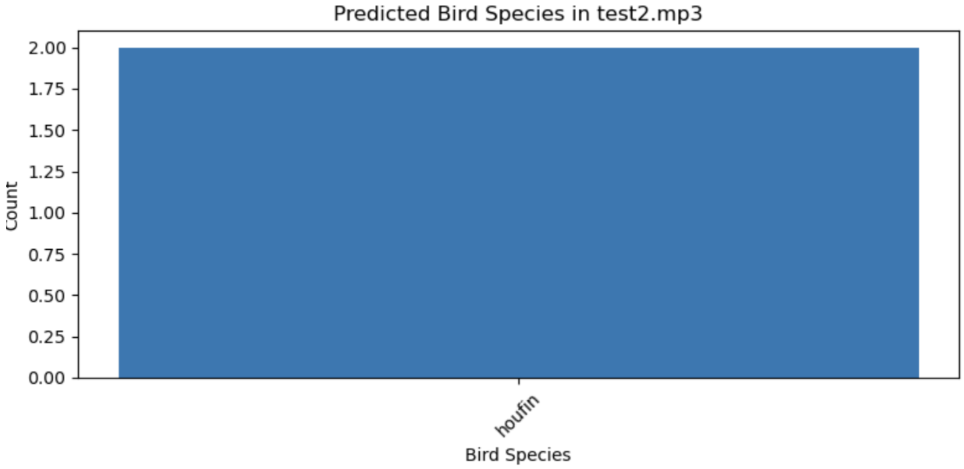


Figure 17. Predicted bird species in Test2.mp3

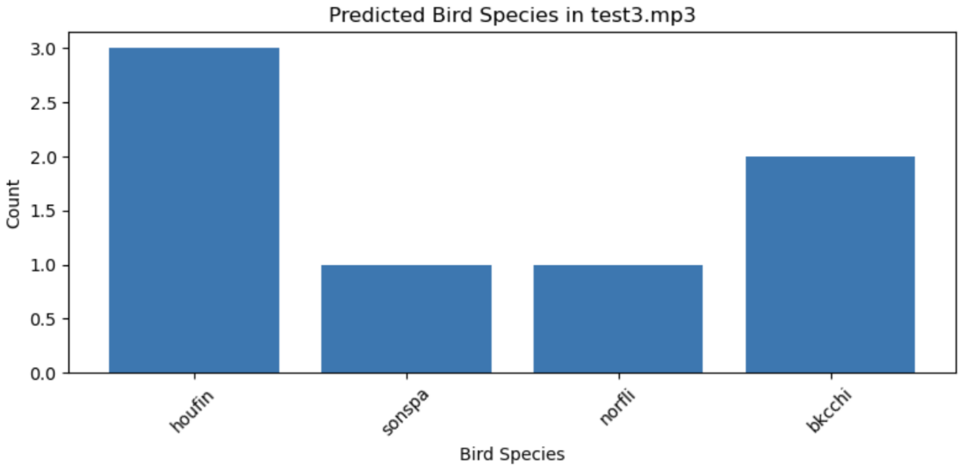


Figure 18. Predicted bird species in Test3.mp3

Our results show that Test1.mp3 and Test3.mp3 are likely to contain multiple bird species, while Test2.mp3 is consistent with a single species call. The top species are in Test1.mp3 and Test2.mp3 is houfin (House Finch) and houfin (House Finch), bkcchi (Black-capped Chickadee) in Test3.mp3.

Discussion

One of the main limitations was the computational time to train the multi-class models. It took approximately 10 minutes to train multi-class model 1, approximately 40 minutes to train multi-class model 2, and approximately over 1 hour to train Tuned Multi-class model 3. In addition, careful processing and validation were required to ensure a balance of bird species in the training and test sets.

Some bird species were more difficult to identify accurately. The first two birds from the 12 birds were selected, the American Robin and the American Crow. The American Crow was difficult to classify in the Binary models, while Bewick's Wren was difficult to classify in Multi-class models. This suggests that the model may struggle with species that have similar acoustic features. Such misclassifications likely stem from the fact that different species can produce calls with overlapping frequency patterns or rhythms, making them hard to distinguish in spectrograms.

We can use decision trees models such as random forest, boosting and support vector machine to classify bird species to perform this task. However, Convolutional Neural Networks (CNNs) work well for this task because they can automatically learn patterns from spectrogram images. Since spectrograms show sound over time and frequency, these models are good at recognizing these visual patterns and differences between bird calls.

Conclusions

In this study, convolutional neural networks were used to classify Seattle's bird species based on spectrograms of their calls. The method entails preprocessing audio data to generate spectrograms, creating and training binary CNN models and multi-class models, and evaluating their performance through classification metrics. Dropout and L2 regularization models were found to be particularly effective in achieving high accuracy and demonstrating robust performance across different datasets, as demonstrated by the results. The CNN model with dropout and L2 regularization emerged as the best performer, achieving 85% accuracy in binary classification and 73% accuracy in multi-class classification tasks. Additionally, some birds were difficult to classify, indicating that further refinement, perhaps including a wider variety of training samples, is needed to improve the model's ability to distinguish between similar calls from a given bird.

In conclusion, binary and multi-class CNN models are a powerful tool for identifying bird species from audio data, with high accuracy and robust performance. In the future, the focus could be on expanding the dataset, fine-tuning the models, and exploring other deep learning architectures to improve the classification accuracy and generalizability of the models.

References

1. James, G., Hastie, T., Witten, D., & Tibshirani, R. (2023). *An introduction to statistical learning with applications in R* (2nd ed.). Springer.
2. Rao, R. (2022). *Xeno-Canto Bird Recordings Extended (A–M) [Data set]*. Kaggle. https://www.kaggle.com/datasets/rohanrao/xeno-canto-bird-recordings-extended-a-m
3. Mendible, A. (2024). *bird_spectrograms.hdf5 [Data set]*. GitHub. https://github.com/mendible/5322/blob/main/Homework%203/bird_spectrograms.hdf5
4. Mendible, A. (2024). *test_birds [Audio files]*. GitHub. https://github.com/mendible/5322/tree/51bd4705bf06d4f46e1848f9261da9b2db3333c0/Homework%203/test_birds
5. Project Jupyter. (2023). *Jupyter Notebook (Version X.X) [Computer software]*. https://jupyter.org
6. Anaconda, Inc. (2023). *Anaconda Distribution (Version X.X) [Computer software]*. https://www.anaconda.com