STATISTICAL MACHINE LEARNING-2

PRACTICAL HOMEWORK 3

SOUNDS OF SEATTLE BIRDS

(NEURAL NETWORKS)

By Alekhya Dabbiru

ABSTRACT

This report evaluates convolutional neural networks for classifying bird songs using spectrograms. Two models were built: one is a binary classification model that distinguishes between Song Sparrow and White crowned Sparrow, and the other is a multi-class classification model developed to identify twelve species.

The binary classifier achieved a test accuracy of 65.75 percent and an area under the curve of 0.6821 after correcting for initial overfitting. For the multi-class model, training for twenty-eight epochs gave better performance than earlier trials, achieving a test accuracy of 41.24 percent.

INTRODUCTION

In this report, bird sounds are classified from recordings to identify species of birds commonly found in the Seattle region. The preprocessed dataset contains MP3 recordings of twelve different bird species. Each species is stored as a separate group in a single HDF5 file. By converting the audio recordings to spectrograms, one can observe how sound frequencies change over time. Because these patterns differ from species to species, they become easier to analyze.

These spectrogram frames serve as the input to our all the classification models. Each frame has been resized and normalized so that pixel values range between zero and one. We use a consistent image size of 128 x 517 pixels to ensure uniformity across all samples. Before training, the frames are shuffled and split into training, validation, and test sets in a 30-30-40 ratio, to maintain a balance of each species in every subset.

The objectives of this study is to develop a binary classifier that distinguishes any two species, in this case, to distinguish between Song Sparrow from White crowned Sparrow, and the next task is to build a multiclass classifier that can identify all twelve target species from their spectrograms, and to evaluate both models on three unlabeled audio clips. Model performance has been measured by accuracy and area under the curve.

THEORETICAL BACKGROUND

Deep Learning: Deep Learning is transforming the way machines understand, learn, and interact with complex data. Deep learning mimics neural networks of the human brain, it enables computers to autonomously uncover patterns and make informed decisions from vast amounts of unstructured data. [1]

Neural Networks: Neural networks are machine learning models that mimic the complex functions of the human brain. These models consist of interconnected nodes or neurons that process data, learn patterns, and enable tasks such as pattern recognition and decision-making.[2]

PROS:

• Parallel processing: Because neural networks are capable of parallel processing by nature, they can process numerous jobs at once, which speeds up and improves the efficiency of computations.

• Non-linearity: Neural networks can model and comprehend complicated relationships in data by virtue of the non-linear activation functions found in neurons, which overcome the drawbacks of linear models. [2]

CONS:

• Overfitting: Overfitting is a phenomenon in which neural networks commit training material to memory rather than identifying patterns in the data. Although regularization approaches help to alleviate this, the problem still exists.

• Computational Intensity: Large neural network training can be a laborious and computationally demanding process that demands a lot of computing power. [2]

APPLICATIONS:

1. Image and Video Recognition: CNNs are extensively used in applications such as facial recognition, autonomous driving, and medical image analysis.

2. Natural Language Processing (NLP): RNNs and transformers power language translation, chatbots, and sentiment analysis.

3. Finance: Predicting stock prices, fraud detection, and risk management. [2]

NEURAL NETWORK'S TERMINOLOGY:

1. Input Layer: This is where the network receives its input data. Each input neuron in the layer corresponds to a feature in the input data.

2. Hidden Layers: These layers perform most of the computational heavy lifting. A neural network can have one or multiple hidden layers. Each layer consists of units (neurons) that transform the inputs into something that the output layer can use.

3. Output Layer: The final layer produces the output of the model. The format of these outputs varies depending on the specific task (e.g., classification, regression). [2]

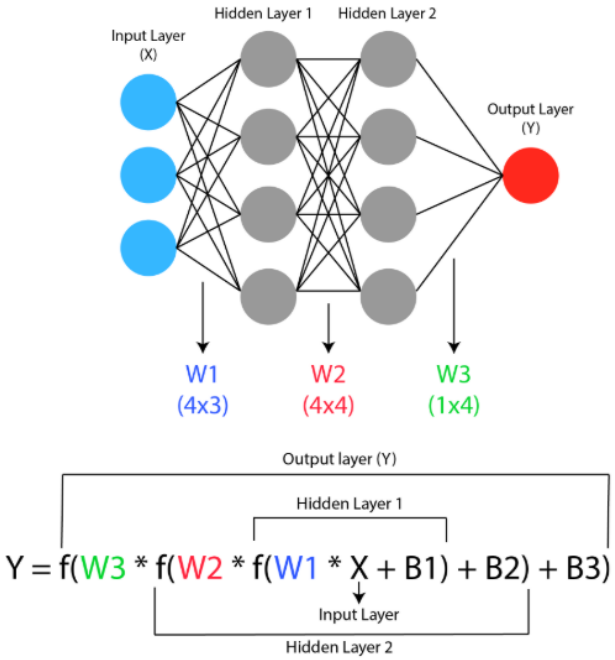

Figure 0: terminology of a neural network; https://circuitlistskipped.z14.web.core.windows.net/simplified-diagram-of-a-neural-network.html.core.windows.net/simplified-diagram-of-a-neural-network.html

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks are a specialized class of neural networks designed to process grid-like data, such as images. They are particularly well suited for image recognition and processing tasks. [3]

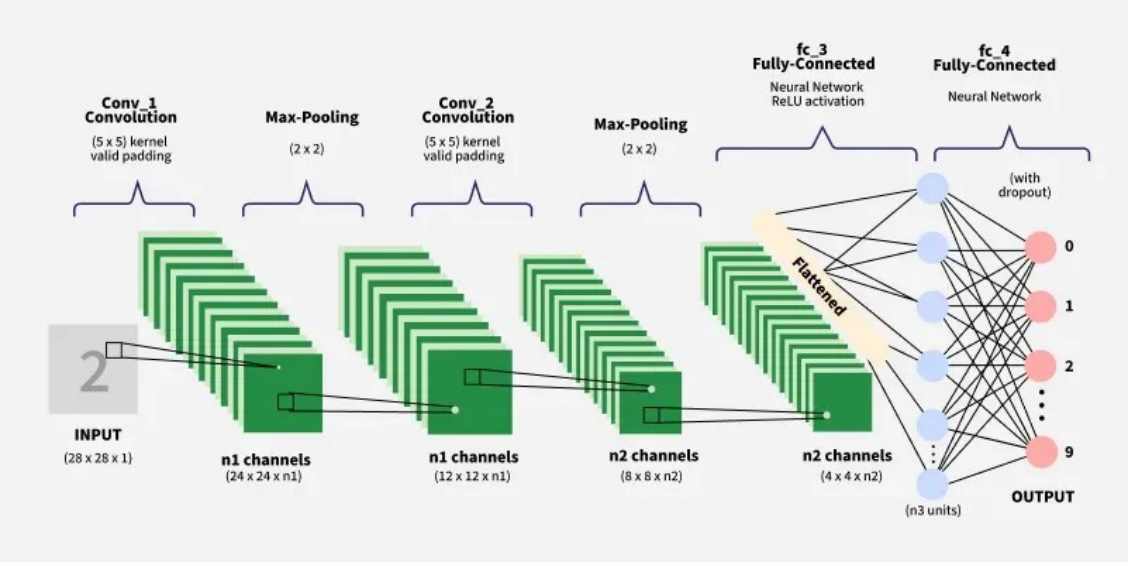

Figure 1 from geeksforgeeks[3]

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks allow the network to “remember” past information by feeding the output from one step into next step. This helps the network understand the context of what has already happened and make better predictions based on that. These incorporate loops that allow information from previous steps to be fed back into the network. This feedback enables RNNs to remember prior inputs making them ideal for tasks where context is important.[4]

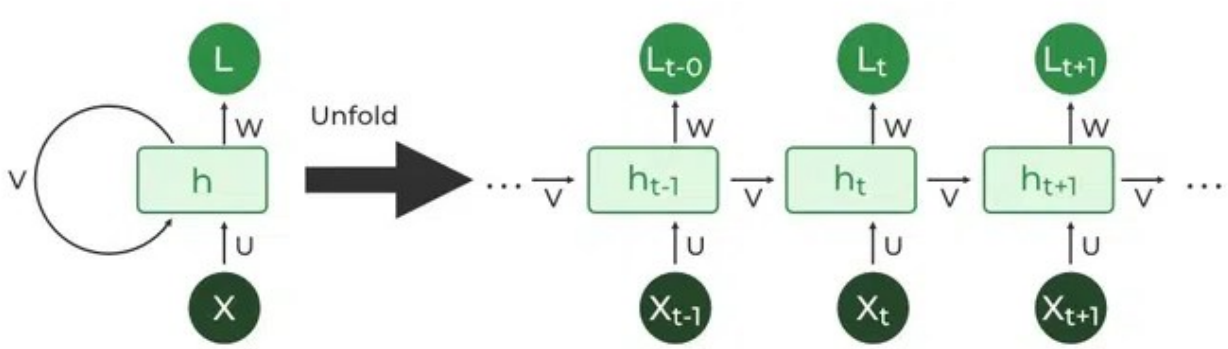

Figure 2: RNN [4]

MULTI-LAYER NEURAL NETWORKS: A multi-layer neural networks consists of an input layer, one or more hidden layers that each apply a linear transformation followed by a nonlinear activation function and an output layer, which contains one or more outputs. This helps in approximating complex functions for classification and regression.

Activation Functions

An activation function is a mathematical function applied to the output of a neuron. It introduces non-linearity into the model, allowing the network to learn and represent complex patterns in the data. Without this non-linearity feature, a neural network would behave like a linear regression model, no matter how many layers it has.[3]

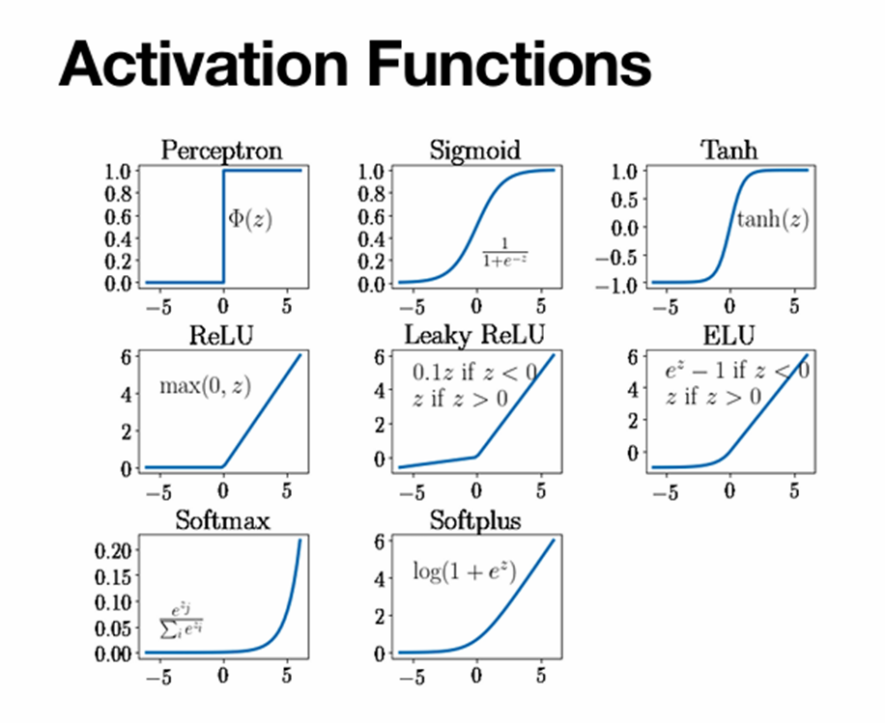

Figure 3:lecture slides

From the above, the simplest is the step function (perceptron), it outputs one whenever its input exceeds zero.

The rectified linear unit (ReLU) passes positive inputs through unchanged and clamps any negative inputs to zero, which tends to speed up convergence during training.

SoftMax function handles multi-class classification problems. It transforms raw output scores from a neural network into probabilities. It works by squashing the output values of each class into the range of 0 to 1, while ensuring that the sum of all probabilities equals 1. [3]

Loss Functions

A loss function is a mathematical way to measure how good or bad a model's predictions are compared to the actual results. It gives a single number that tells us how far off the predictions are. The smaller the number, the better the model is doing. Loss functions are used to train models.[5]

Binary Cross-Entropy Loss is also known as Log Loss and is used for binary classification problems. It measures the performance of a classification model whose output is a probability value between 0 and 1.[5]

$$\text{Binary Cross-Entropy} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Figure 4: binary cross entropy [5]

Categorical Cross-Entropy Loss is used for multiclass classification problems. It measures the performance of a classification model whose output is a probability distribution over multiple classes.[5]

$$\text{Categorical Cross-Entropy} = -\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij} \log(\hat{y}_{ij})$$

Figure 5: categorical cross entropy[5]

SPECTROGRAM: A spectrogram shows the signal's frequency content evolves over time by applying the Fourier transform to overlapping short windows and mapping each window's magnitude to a color or brightness scale.

METHODOLOGY

The dataset is in a HDF5 file, with each of the twelve bird species stored in its own species group. Spectrograms are created by converting MP3 clips into time frequency images. Each image is scaled so pixel values lie between zero and one and resized to 128 x 517 pixels. Next, species labels are mapped to integer codes to facilitate model training. Finally, the full dataset is shuffled and split into training (sixty percent) and testing (forty percent) sets, ensuring that each species remains proportionally represented.

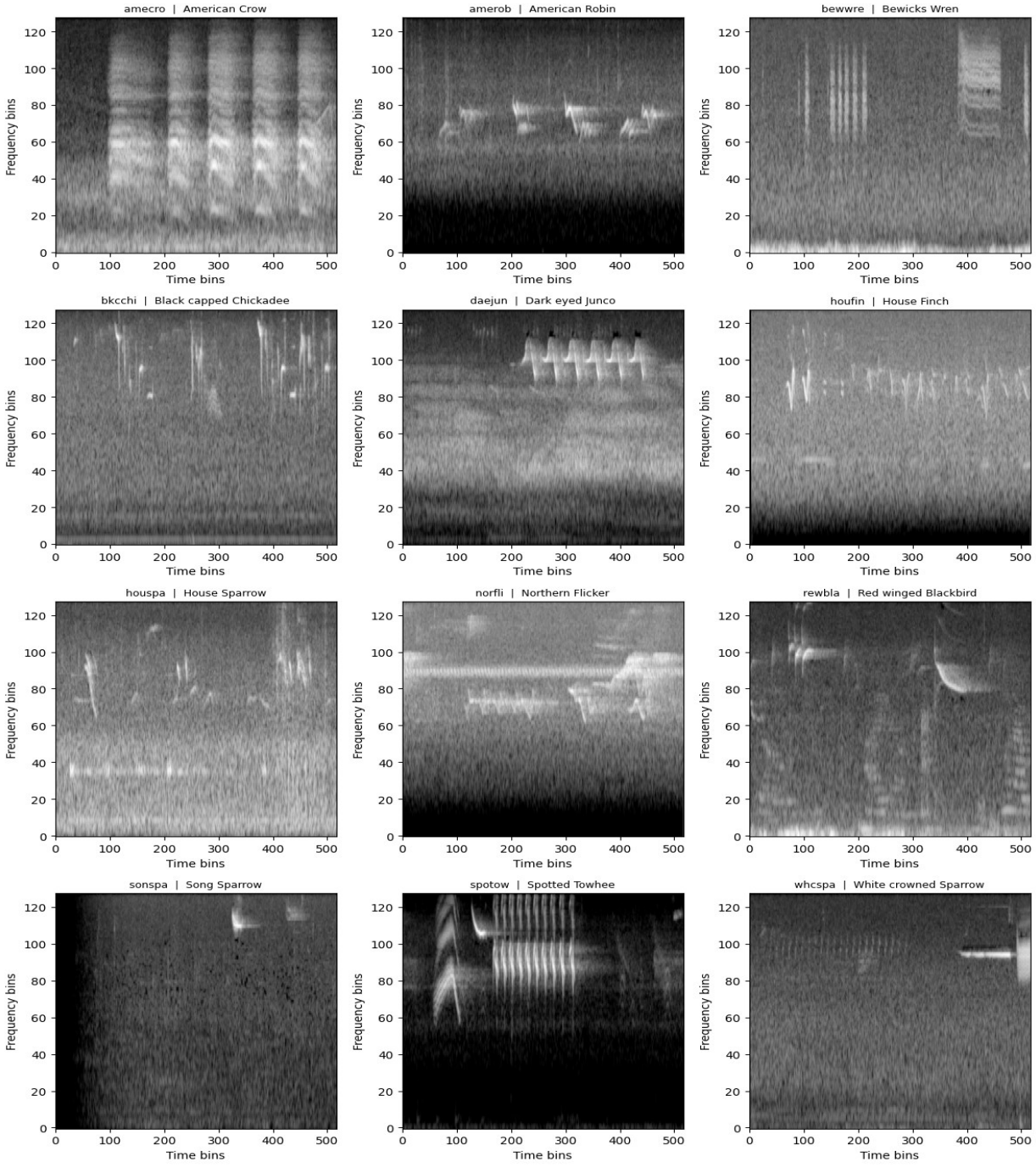

Figure 6: from code

DATA PREPARATION:

For the last part, three audio files called “test1,” “test2,” and “test3” are loaded at 22050 Hz. Each file’s waveform is shown so its shape and volume are checked.

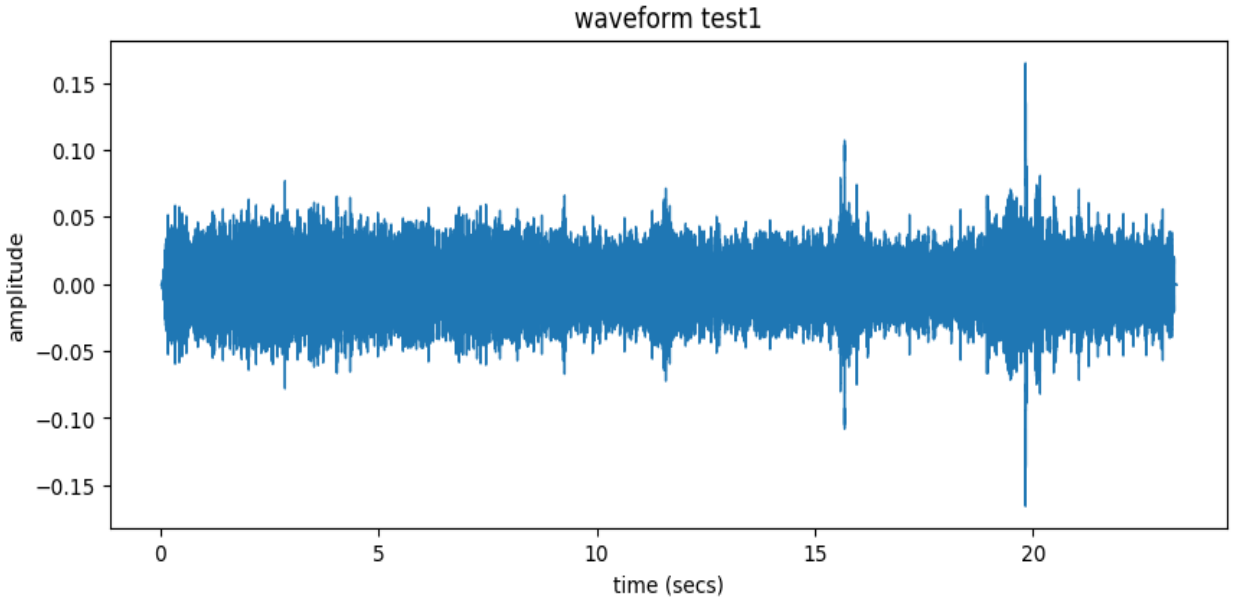

Figure 7: waveform1

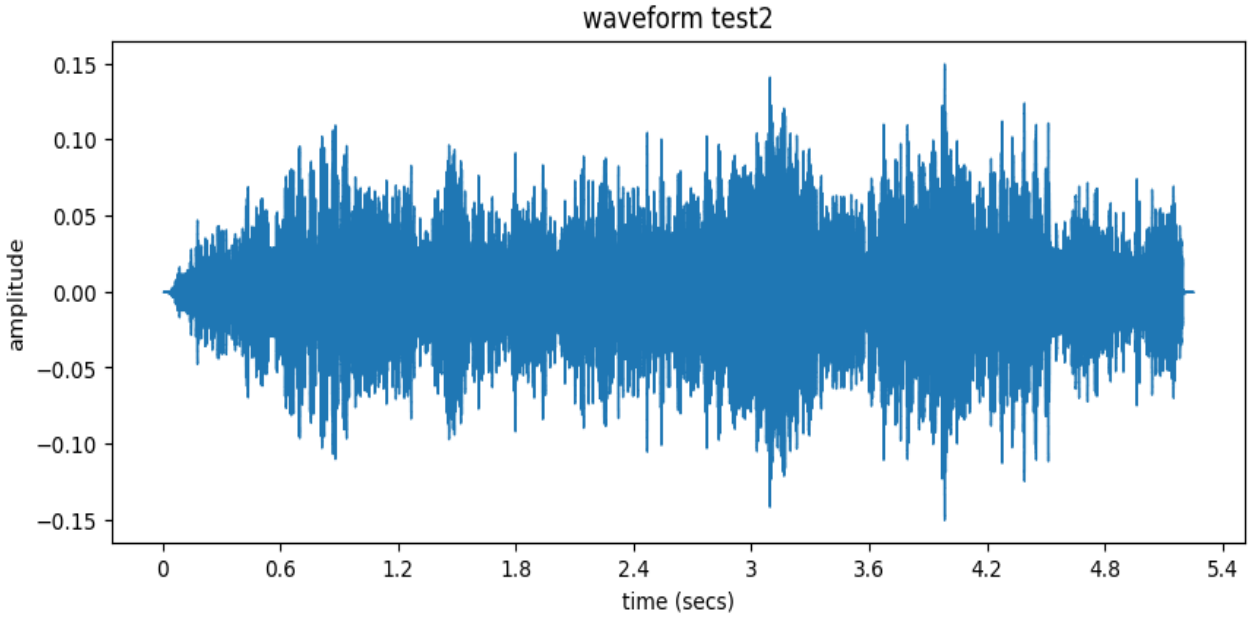

Figure 8: waveform2

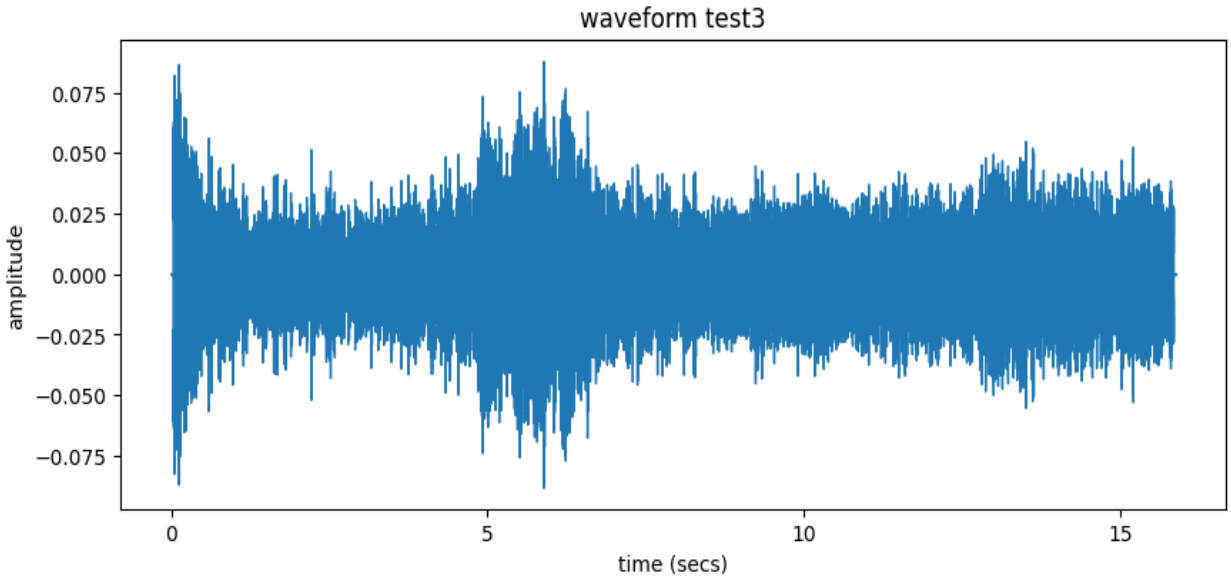

Figure 9: waveform 3

Then, these clips are turned into Mel scale spectrogram images by running a fast Fourier transform with 2048 points, stepping every 512 samples, and using 256 frequency bands. The power values are converted to decibels and displayed as pictures with time on the bottom, frequency on the side, and a color scale for loudness. Finally, each spectrogram is saved in its own dataset inside an HDF5 file. So, there are three such. This makes it easy to load the prepared test data later without having to recreate the images.

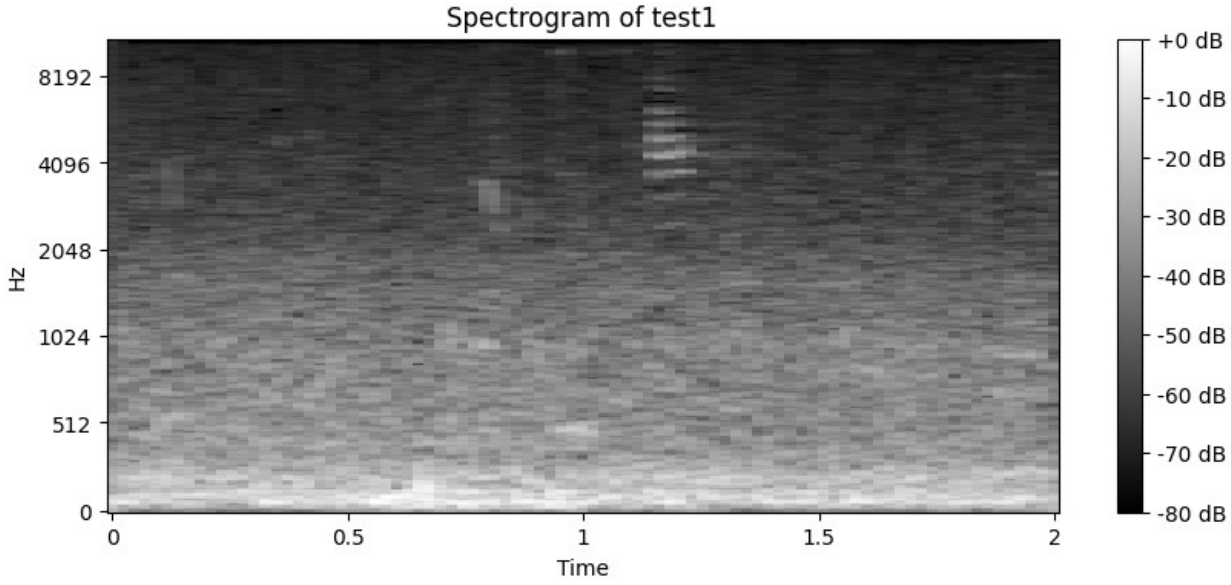

Figure 10: spectrogram 1

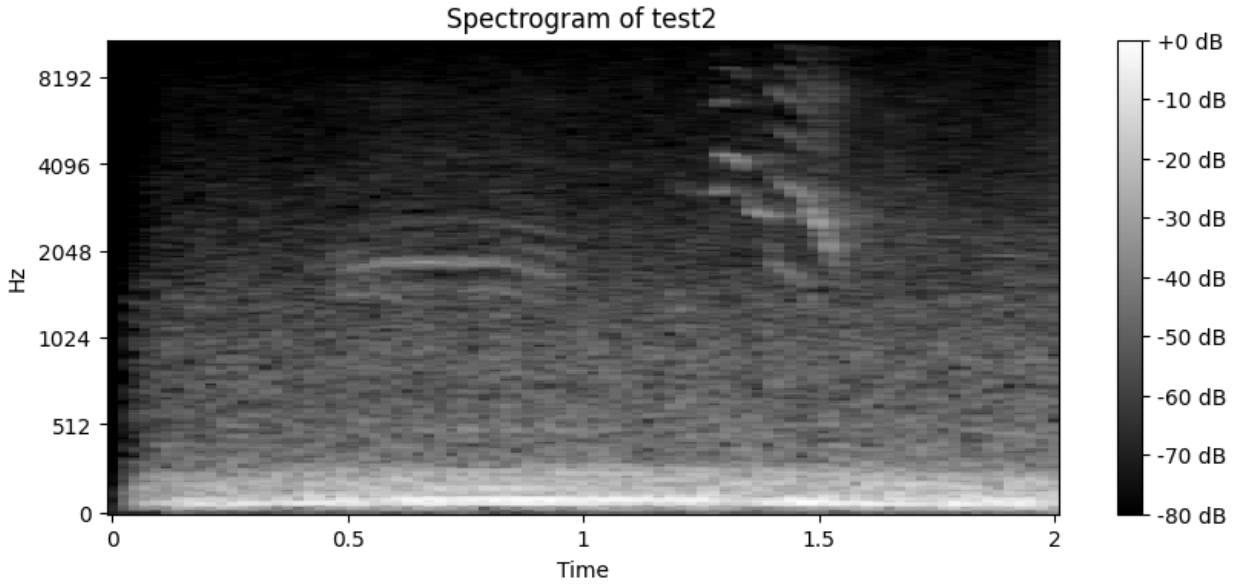

Figure 11: spectrogram2

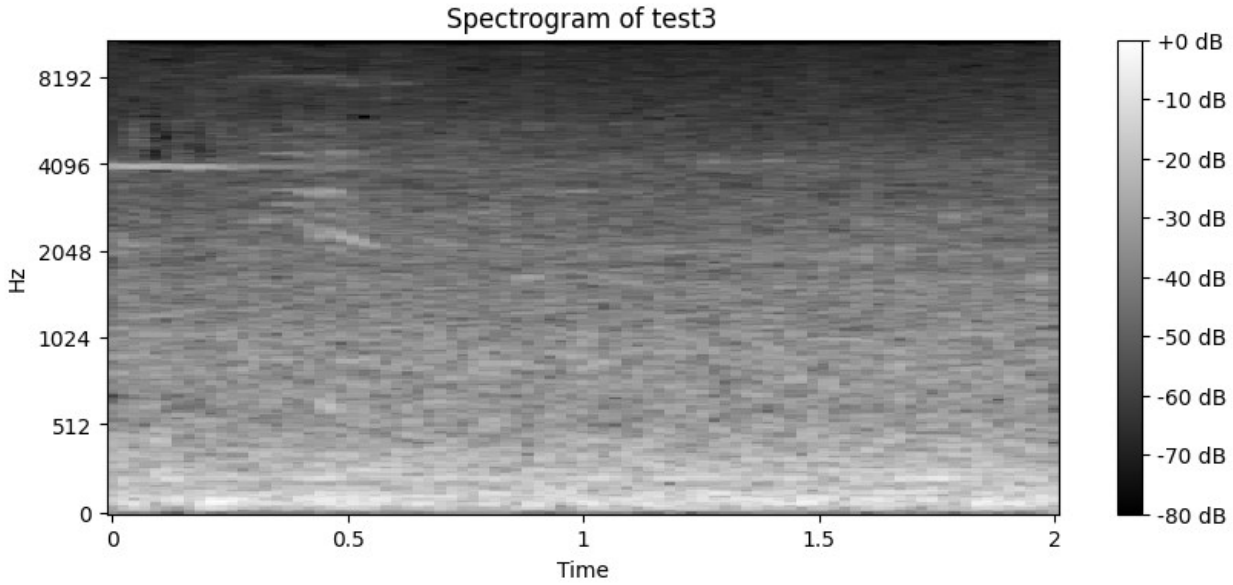

Figure 12: spectrogram 3

BINARY MODEL:

The binary model was designed to evaluate Song Sparrow and White crowned Sparrow using their spectrogram images. At the start, working with the h5py library was unfamiliar, so loading the data required several rounds of debugging.

Initial training without any validation checks produced one hundred percent accuracy on the training set, a clear sign of overfitting. After learning about this issue in class, early stopping was added to the training loop. Training now stopped automatically when the validation loss did not improve for ten epochs, and the model reverted to the weights that gave the best validation score. With this change and model checkpointing, the binary classifier achieved a test accuracy of 65.75 percent and an area under the curve of 0.6821.

The network has used two layers of convolution followed by pooling .A small fully connected layer with a sigmoid output produced the final probability. The Adam optimizer and cross entropy loss guided training.

MULTI-CLASS CLASSIFICATION:

The multiclass model built on the binary design by adding another convolution layer. Each convolution layer was followed by a pooling step to reduce the image size and highlight important features. After these layers, a dense layer with softmax function produced a probability for each of the twelve species. Data loading continued to use the same HDF5 pipeline once the initial issues were resolved.

Many versions of the multiclass architecture were tried, changing the number of filters, layer depths, and dropout rates. Despite these experiments, the final configuration proved to be the better one. During training, the model often reached over 70 percent accuracy on the training data but struggled to exceed 40 percent on the validation set. To prevent overfitting, training stopped automatically when validation loss failed to improve for ten rounds, and the best model weights were saved for testing and evaluation.

The confusion matrix for the multiclass model showed a clear bias toward species with the most examples and against those with fewer or very similar calls. Some rows and columns stood out as unexpectedly strong or weak, reflecting both class imbalance and acoustic overlap among certain species. Although this pattern looks unusual, it highlights where the model makes consistent errors

COMPUTATION RESULTS:

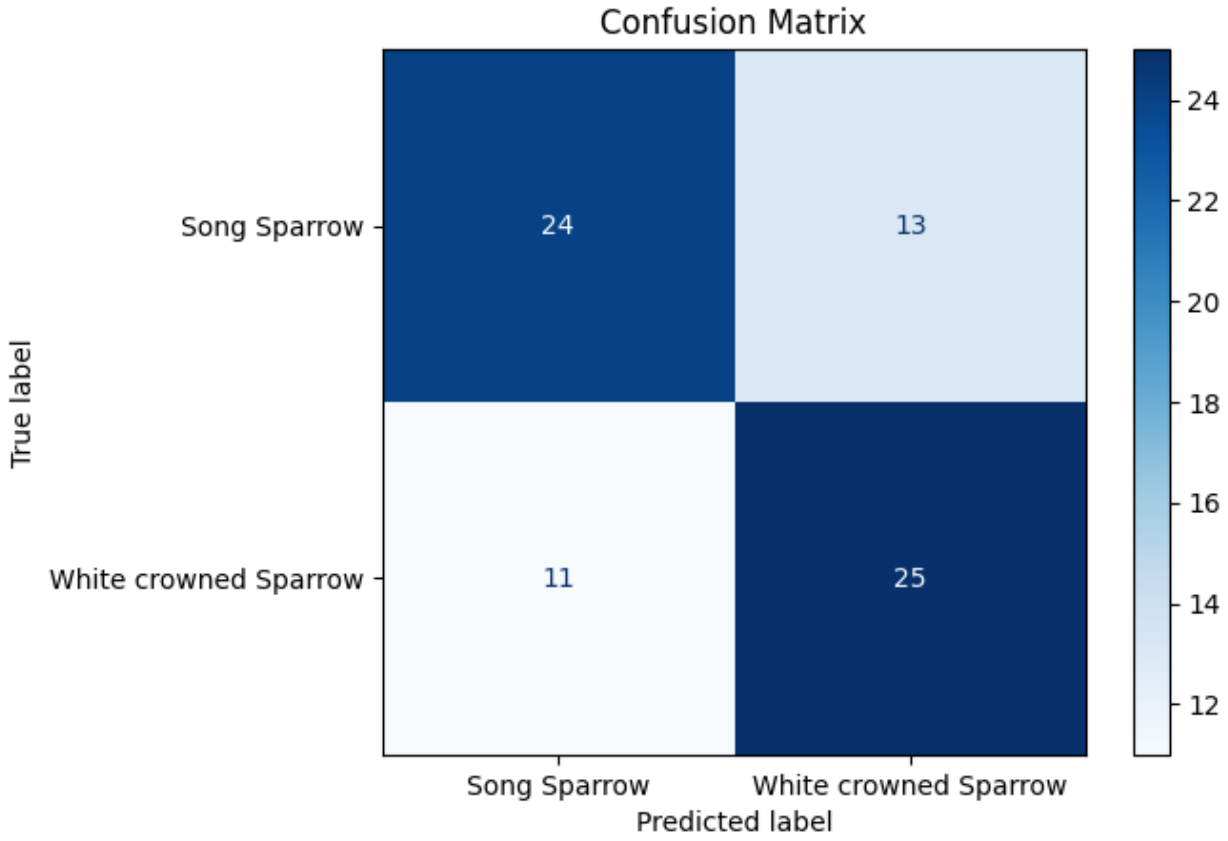

Figure 13: confusion matrix

The confusion matrix above shows that out of 37 Song Sparrow samples, 24 were correctly identified and 13 were mistaken for White crowned Sparrow. For White crowned Sparrow, 25 out of 36 samples were correctly classified while 11 were mislabelled as Song Sparrow. This yields an overall test accuracy of 67.2%. The pattern of errors confirms that distinguishing these two sparrows remains challenging, but the model performs reliably once early stopping and checkpointing are in place.

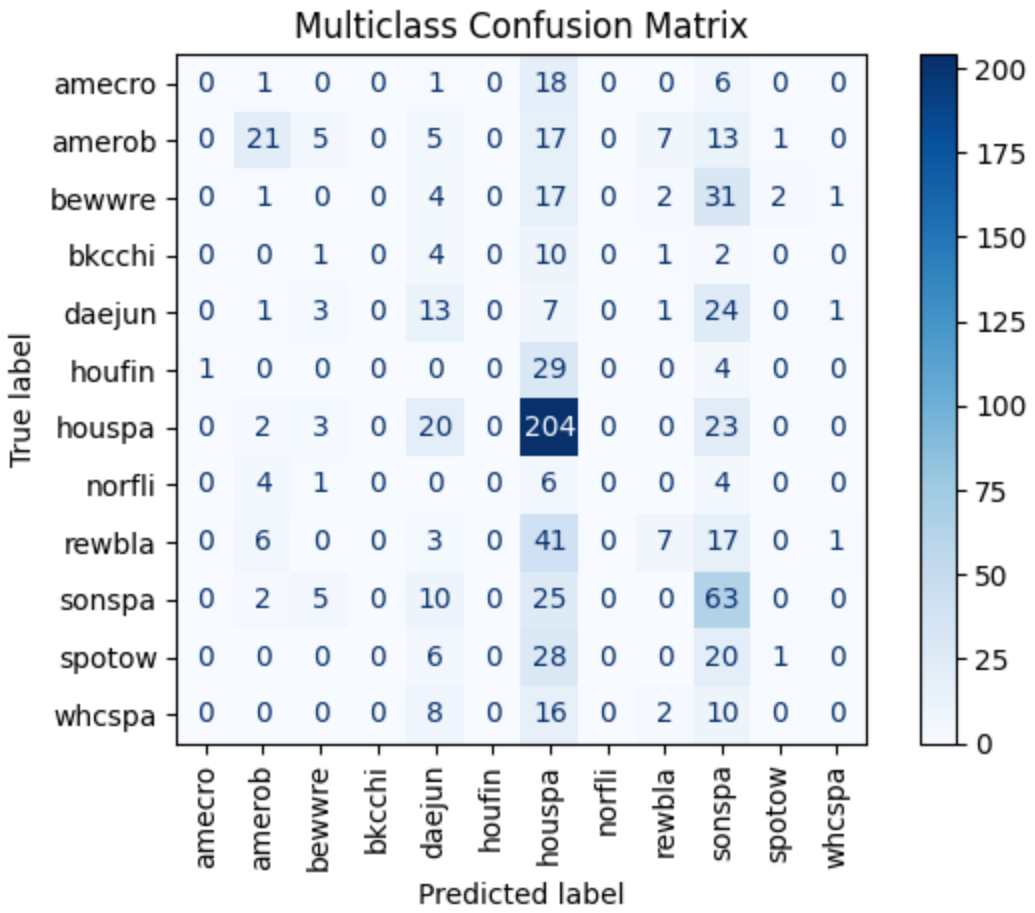

Figure 14: multi-class confusion matrix

The multiclass confusion matrix and classification report reveal how the model handles all twelve species. It is evident from the above confusion matrix that, the model predicts House Sparrow most often, getting 204 right but also miscalling many other species as House Sparrow. Some species with fewer examples, like Northern Flicker and Black Capped Chickadee, are often mistaken for others. Song Sparrow is correctly identified 63 times but still confused with House Sparrow on 25 occasions.

House Sparrow achieves the highest recall of 81 percent and a precision of 49 percent, resulting in an F1 score of 60 percent. Song Sparrow follows with a recall of 60 percent and precision of 29%.

	Class	Precision	Recall	F1 Score
0	amecro	0.000000	0.000000	0.000000
1	amerob	0.552632	0.304348	0.392523
2	bewwre	0.000000	0.000000	0.000000
3	bkcchi	0.000000	0.000000	0.000000
4	daejun	0.175676	0.260000	0.209677
5	houfin	0.000000	0.000000	0.000000
6	houspa	0.488038	0.809524	0.608955
7	norfli	0.000000	0.000000	0.000000
8	rewbla	0.350000	0.093333	0.147368
9	sonspa	0.290323	0.600000	0.391304
10	spotow	0.250000	0.018182	0.033898
11	whcspa	0.000000	0.000000	0.000000

Figure 15: bird species evaluation metrics

Many species show zero precision and recall, reflecting either severe class imbalance or high similarity to other calls. Overall, the model favors the most abundant classes and struggles to learn rare or similar species.

```
Epoch 1/5
62/62 [==============================] - 97s 2s/step - loss: 2.2666 - accuracy: 0.3029
Epoch 2/5
62/62 [==============================] - 88s 1s/step - loss: 2.2182 - accuracy: 0.3165
Epoch 3/5
62/62 [==============================] - 95s 2s/step - loss: 2.1717 - accuracy: 0.3150
Epoch 4/5
62/62 [==============================] - 89s 1s/step - loss: 2.1132 - accuracy: 0.3246
Epoch 5/5
62/62 [==============================] - 89s 1s/step - loss: 2.0288 - accuracy: 0.3523
Test Spectrogram 1 predicted as sonspa
Test Spectrogram 2 predicted as sonspa
Test Spectrogram 3 predicted as sonspa
```

Figure 16: task 3

When applied to the three unlabeled test clips, the multiclass model predicted Song Sparrow for all three. This bias is likely because Song Sparrow had the most examples in the training data and also likely due to some background sounds. To fix this issue, the dataset needs to be more balanced.

DISCUSSION

Initially, working with the audio data and h5py library was a new experience and felt unusual. It took time to explore the file structure and understand the dataset keys. At times, the amount of experimentation and debugging felt overwhelming.

Several trials were run to choose the best binary model. The first trial trained for one hundred epochs with a batch size of 32 and took around two minutes. It reached a test accuracy of 62 percent and an AUC of 0.70. The second try had two hundred epochs but stopped early at epoch 137 when validation loss stopped improving. This try has run nearly less than eight minutes and yielded similar accuracy. The third and final trial returned to one hundred epochs and ran for about three and a half minutes. This run produced the best balance between training and validation accuracy, with a test accuracy of 66 percent and an AUC of 0.69. The final model was selected by comparing accuracy and loss plots from all three trials and choosing the one that is better.

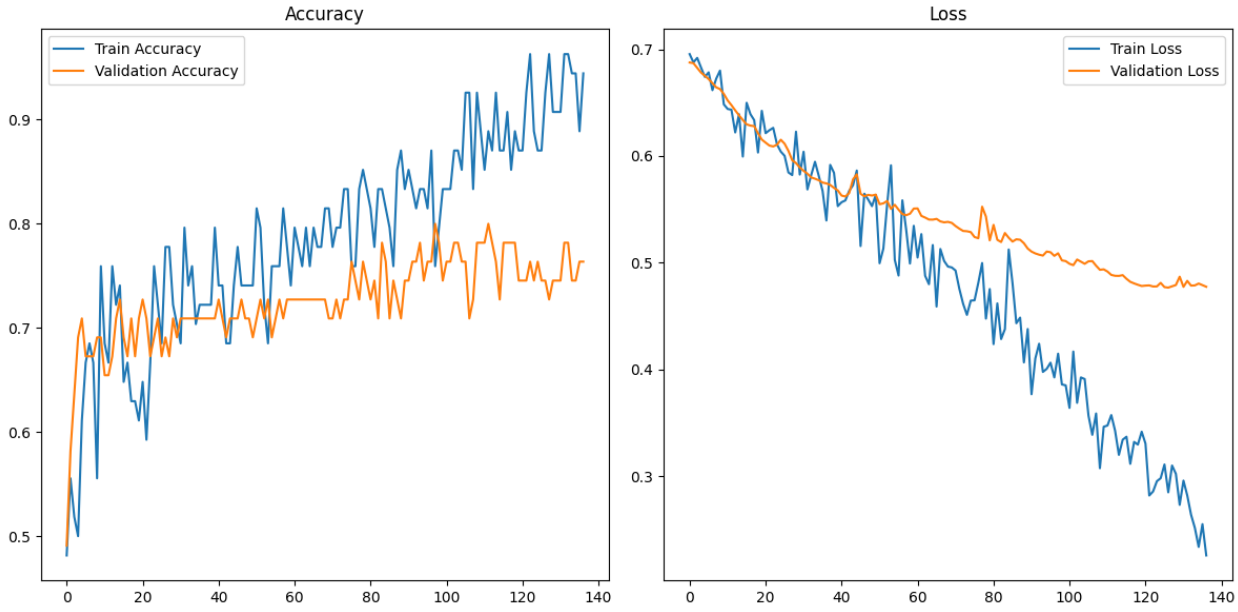

Figure 17: accuracy and loss plot of binary model

For the multiclass model, many attempts were tried, all using the same network design, batch size of 32, and learning rate of (1e-4). The first run (which is also the selected final run for this model) has stopped at 51 epochs after about 24 minutes of training and achieved 31 percent test accuracy. A second trial has run until 95 epochs in about 33 minutes and has got similar accuracy to the previous one. A third run stopped at 178 epochs, ran for 40 minutes and dropped to 17 percent accuracy. A fourth attempt ended at 157 epochs for roughly around 48 minutes and dropped more to eight percent. None of

the longer runs improved performance beyond the first selected run.

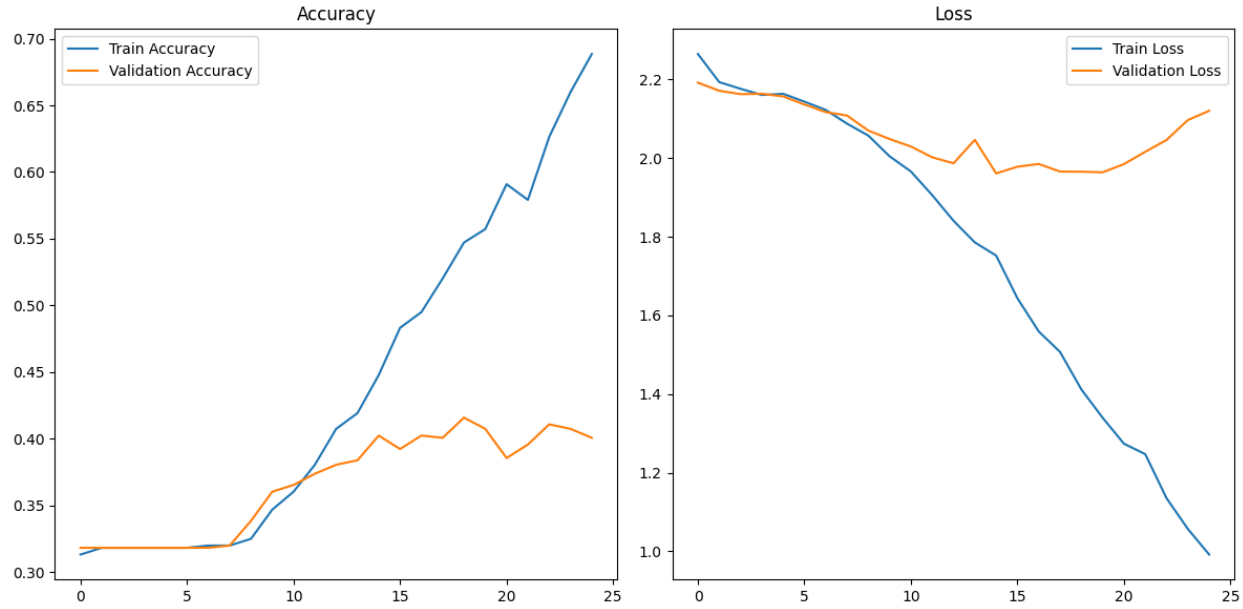

Figure 18: accuracy and loss plot of multi-class model

The class imbalance played a major role in the results. Some species were much harder for the multiclass model to predict correctly. House Sparrow appeared as the predicted label more than any other bird, even when it was wrong. This happened because there were more House Sparrow samples in the training data, and its call can also sound like background noise.

Song Sparrow and White-crowned Sparrow were also often mixed up. They have similar notes and similar-looking spectrograms. Without a clear difference, the model could not tell them apart.

Northern Flicker and Black-capped Chickadee had the lowest correct prediction rates. Their calls are faint and often lost in the noise. These overlaps in sound patterns made it hard for the network to learn a unique signature for each bird.

The top 3 external clip predictions

	clip	top1_species	top1_probability	top2_species	top2_probabaility	top3_species	top3_probabality
0	test1	House Sparrow	0.805277	American Robin	0.126353	Song Sparrow	0.023167
1	test2	House Sparrow	0.643778	American Robin	0.204123	Song Sparrow	0.053612
2	test3	House Sparrow	0.853333	American Robin	0.108637	Song Sparrow	0.013637

Figure 19: Task 3

All three test clips were identified as House Sparrow with high confidence, over 80 percent. The next most likely bird was American Robin, but its probability was very low around 20 percent, even Song Sparrow (is third) has below five percent probability. The large difference between the first prediction and the next two show that the model is sure that these clips contain House Sparrow calls.

Other than neural networks, using simple classifiers can also be used. Support vector machines, or logistic regression can be used by converting each spectrogram into a list containing numbers. Random forest or gradient boost trees can also be used if the spectrograms are converted into a set of features, like average frequency and bandwidth. These features will be the input for these tree models, and prediction can be made.

I think central neural networks are best for this case because spectrograms are like pictures. Convolutional networks learn to find patterns in pictures by themselves. And there is no need to manually design the features. The network can learn patterns that detect trills, harmonics, and other sound shapes over time and frequency directly from the data. Since, we have a good number of spectrograms, the network can learn these patterns well and give better results than simple models.

CONCLUSION

The binary model predicted Song Sparrow versus White crowned Sparrow calls with about 66 percent accuracy and an AUC of 0.68, while extending the same approach to the multi-class model of twelve species, the accuracy was around 41 percent. So, the models struggled because many songs sound alike, and the model overfit to the common species. Overall, the study highlights the potential and the limits of neural networks for bird song identification, pointing to areas for further improvement in data preprocessing, model training and handling of complex audio data.

REFERENCES

[1]. Introduction to Deep Learning | GeeksforGeeks

[2]. What is a Neural Network? | GeeksforGeeks

[3]. Convolutional Neural Network (CNN) in Machine Learning | GeeksforGeeks

[4]. Introduction to Recurrent Neural Networks | GeeksforGeeks

[5]. Loss Functions in Deep Learning | GeeksforGeeks

[6]. Quick Start Guide — h5py 3.13.0 documentation

[7]. How to convert MP3 to WAV in Python - Stack Overflow

[8]. Audio classification using spectrograms | GeeksforGeeks

[9]. Lecture Presentation Slides, Deep Learning, Deep Learning II, DATA 5322, Seattle University, 2025.