Deep Learning for Bird Sound Recognition

Abstract

Precise recognition of bird species through audio relies a lot on careful attention to audio tracks from multiple species which consumes a lot of time and induces human error. To solve this I utilized data from Birdcall Competition data from 2020 originally from xeno-canto. I approached the problem in two stages, first a binary classification to distinguish between the two most frequently occurring species (highest number of samples in the dataset) and then a multi-class classification across all 12 species. Since there was a class imbalnce i downsampled to match the lowest number of samples for bird species. I also used focal loss, data augmentation and trained CNN architecture from scratch which resulted in 80% and 35% accuracies for binary and multi-class respectively.

Introduction

Xeno-canto consists of wildlife sounds around the world. For this report, I focused on recognition of 12 Bird Sounds common in the Seattle area. The 12 species are American crow(amecro), American robin(amerob), Bewick's wren(bewwre), Black-chapped Chickadee(bkcchi), Dark-eyed Junco(daejun), House Finch(houfin), House Sparrow(houspa), Northern Flicker(norfli), Red-winged Blackbird(rewbla), Song sparrow(sonspa), Spotted Towhee(soptow) and White-crowned Sparrow(whcspa).

This report uses different Neural Networks architecture for binary and multi-class and shows performance based on hyperaparam tuning, dataset augmentation and use of loss function.

Theoretical Background

A Neural Network is a machine learning model which learns patterns from data which is sort of a simplified version of how the brain works. It takes in information, processes it through several layers and then makes predictions on what it has learned.
It consists of 3 layers:

1. Input
   This layer is the way the network sees the data - it can be numbers, text , images and in our case spectrograms.

2. Hidden
   This is where learning takes place. Each layer takes in information, does some math with adjustable values called weights and decides which features are important. Activation function is used to help the network handle complex patterns - not just straight lines or simple relationships.

3. Output
   This is where the network gives its final output. For Ex In our case it would say “I am sure 85% it's a House Sparrow”.

4. Learning Process
   When the network gets something wrong, it uses that mistake to adjust its internal settings. This process is called backpropagation and is like giving feedback so it can improve each time.

Neural Networks working through an example:

In Fig.1, Nodes X1, X2 and X3 are the input features or values (in this spectrogram images).

Each input is connected to every neuron in the hidden layer with a weight.

A1, A2, A3 and A4 make up the hidden layer, Each neuron receives a weighted sum of all input nodes :

$$A_j = f(W_{j1}X_1 + W_{j2}X_2 + W_{j3}X_3 + b_j)$$

Where W is weights, b is bias and f is an activation function(Ex ReLu)
Hidden layers introduce non-linearity to the network.

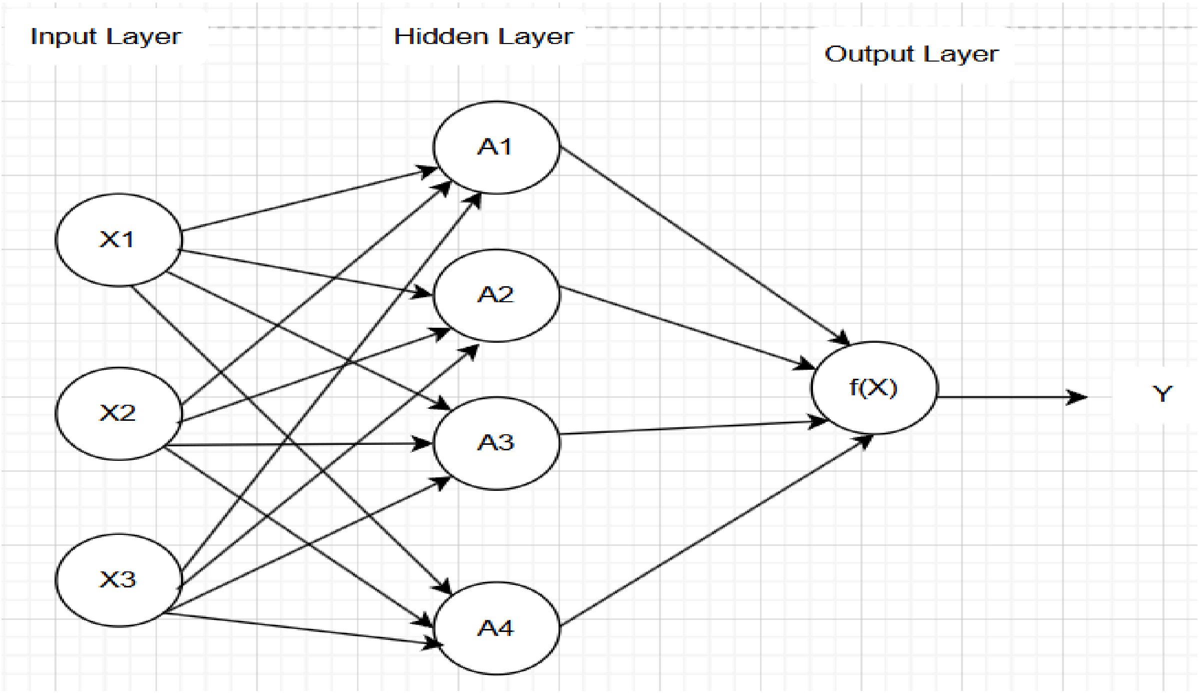


Fig.1 Graphical Example of Feed forward Neural Network

f(X) produces output Y. It takes all input from hidden nodes(A1 to A4) and computes

$$Y = f(W_1A_1 + W_2A_2 + W_3A_3 + W_3A_3 + b)$$

If there is more than 1 hidden layer it is called Multi-layer Neural Network.
One of the most popular uses was image classification and was popularized by CNN(Convolution Neural Network).

CNNs are built using two main types of hidden layers: convolution layers and pooling layers. The convolution layers are responsible for detecting patterns in images by scanning them with filters. Each filter slides over the image, focusing on small patches at a time, multiplying its values with the image patch and summing them up. When the pattern in the patch match closely matches what the filter is looking for - like an edge or texture - the output is a high value, indicating a strong match.

After these patterns are detected, pooling layers help simplify the data by reducing the image size while keeping the most important information.
For example, max pooling selects the highest value from each region, effectively summarizing features while making the network faster and more efficient.

Methodology

The Audio clips were first converted to spectrograms. The data is preprocessed the following way:

- The sound clip is subsampled to 22500Hz.
- The first 3 sec is selected.
- A spectrogram is produced for each 2-second window, resulting in a 128 (frequency) x 517 (time) "image" of the bird call.
- All bird calls for all clips in a given species are saved individually (stored in HDf5 format).

Binary Classification

For the Binary Classification task, I focused on House Sparrow and Song Sparrow, labeling them as 0 and 1 respectively. The data was converted to tensors and then split into 70% for training, 15% for validation and 15% for testing.
I designed a custom CNN architecture that included dropout and used a weighted loss function to address class imbalance. The model was trained using CrossEntropy Loss and Adam Optimizer with early stopping based on validation loss.
Adam is a popular Optimizer that helps models learn faster by adjusting how much weight changes during training whereas Cross Entropy is a loss function that tells the model how far off its predictions are from actual labels. It works by comparing the predicted probabilities to the true class and gives higher penalties when the model is more confident about a wrong answer.

MultiClass Classification

I trained CNN from scratch to classify 12 bird species based on spectrogram inputs. To ensure the model learned equally across all classes, the dataset was downsampled to match the number of samples in the least-represented species, creating a balanced training set. The architecture was enhanced by introducing batch normalization to fasten the training and using adaptive average pooling instead of flattening to reduce sensitivity to the spatial layout.

To save memory and to increase diversity in the dataset, I applied on-the-fly augmentation using time-masking and frequency masking. These techniques randomly hide parts of the spectrogram along the time or frequency axis, inducing variations in bird calls, such as when a bird call is partially completed or overlaps with background noise.

I used focal loss to help the model focus more on hard-to-classify samples, which is helpful in multi classes when some species are more similar to each other. Training was done using Adam Optimizer with early stopping based on validation loss.

Results

Binary Classification

House sparrow (labelled as 0), song sparrow(labelled as 1) had 630 and 263 samples respectively.

No.	Key Features	Loss Function	Accuracy	Comments
1	Basic 2-layer CNN	CrossEntropyLoss	~71%	Overfitting No dropout
2	CNN + Dropout + L2 regularization	CrossEntropyLoss	~66%	Better Regularization; Underfit
3	Dropout + Class Weighting	Weighted CrossEntropyLoss	~67%	Slight improvements; Early Stopping
4	BatchNorm + Class Weights + Adaptive Pool	Weighted CrossEntropyLoss	~80%	Best Performance + stable training

Table 1: Binary Classification Models comparison

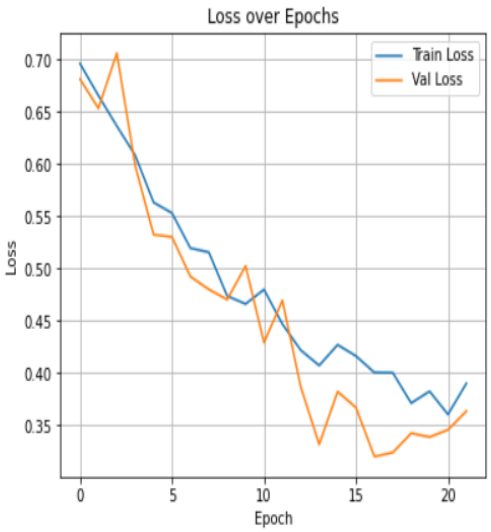


Fig. 2 Loss over Epochs (Binary)

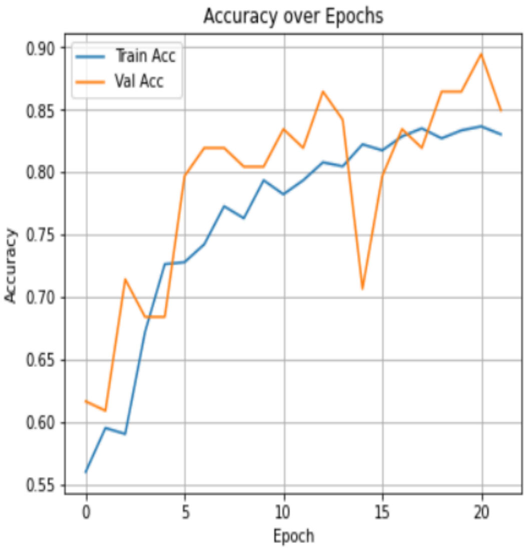


Fig. 3 Accuracy over Epochs (Binary)

Both Fig. 2 and Fig. 3 are for Attempt 4.

In Fig. 2, the graph shows a steady drop in both training and validation loss, meaning the model was learning consistently over time. Validation loss even dips below training loss towards the end, which is a good sign as the model is generalizing well and not just memorizing the training data.

In Fig. 3, the graph shows a steady rise in both training and validation accuracy, indicating that the model was consistently improving its predictions. Validation accuracy remains slightly higher than training accuracy showing no signs of overfitting.

Fig. 4 shows classification report for Attempt 4. This is the best performing model with an overall accuracy of 81%. It was especially strong at recognizing Class 0, predicting it with high precision and recall. Class 1 was a bit more challenging as the model caught most of these cases (recall of 0.71) but it also had a few false positives, leading to a lower precision of 0.53, Still the overall balance is the best compared to previous attempts with a macro f1 score of 0.74.

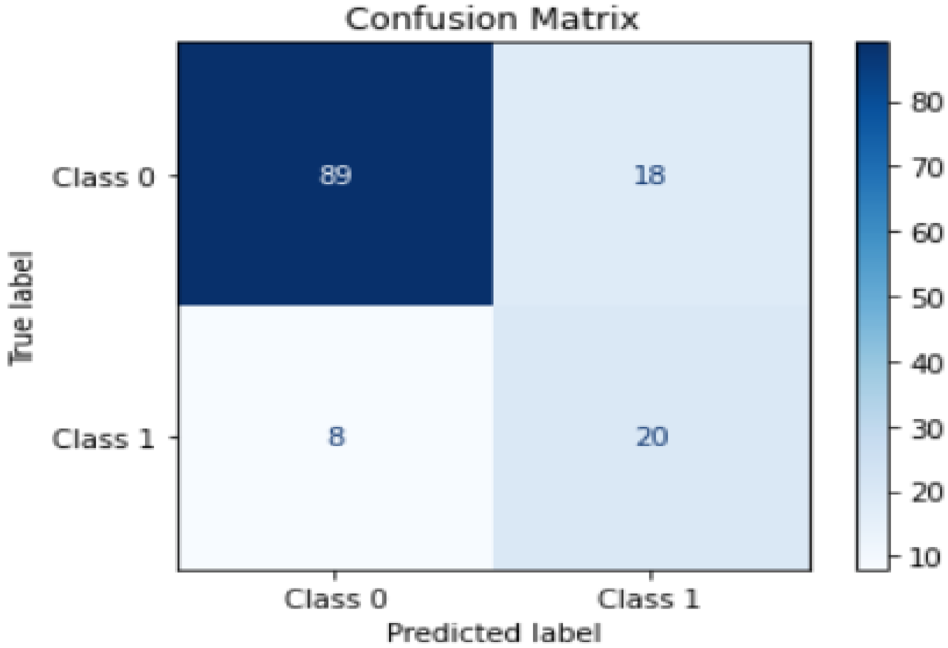


Fig. 4 Confusion matrix (Binary)

MultiClass Classification

The Number of samples for each species is shown below:

Species	No. of samples
American crow	66
Bewick’s wren	144
Black-chapped Chickadee	45
Dark-eyed Junco	125
House Finch	84
House Sparrow	630
Northern Flicker	37
Red-winged Blackbird	187
Song sparrow	263
Spotted Towhee	137
White-crowned Sparrow	91
American robin	172

From the above table we can infer that there is class imbalance where Northern Flicker has least samples(37) and House Sparrow has the highest(630).

No.	Key Features	Dataset	Loss Function	Accuracy	Comments
1	Baseline CNN + Class Weights	Imbalanced	Weighted CrossEntropy	17.1%	Model biased towards frequent class
2	Depper CNN +Focal	Imbalanced	Focal Loss	9.1%	Overfitted

	Loss + Weighted Simplifier				
3	Downsampled + Focal Loss	Balanced (Train only)	Focal Loss	18-19%	More stable but not generalized for all classes
4	Downsampled + Focal Loss + Separate Test Set	Balanced (Train + Test)	Focal Loss	35%	Best generalizations across all the models

Table 2 : Multi-Class Classification Models comparison

Across the four MultiClass classification attempts, performance improved as class imbalance was addressed and data handling was refined. Starting with Attempt 2, all models incorporated on the fly augmentation using time and frequency masking.
The baseline model in Attempt 1 performed reasonably well but showed a clear bias toward majority class.

In Attempt 2, we introduced Focal Loss and downsampled the dataset, but this led to unstable training and poor performance, likely due to overcompensation and early stopping. Attempts 3 and 4 both used balanced datasets; Attempt 3 showed improved training stability but struggled to generalize across all classes. In contrast, Attempt 4, which used a balanced dataset along with a separate held-out-test set, achieved the best overall accuracy of 35%.

In both Binary and MultiClass setup early stopping was included based on validation loss.

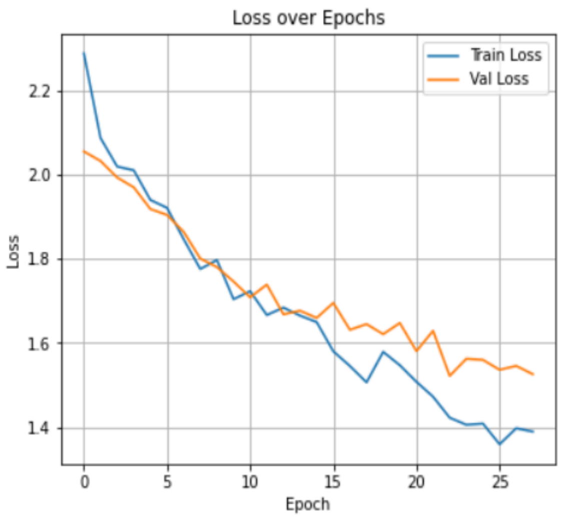


Fig. 5 Loss over Epochs (MultiClass)

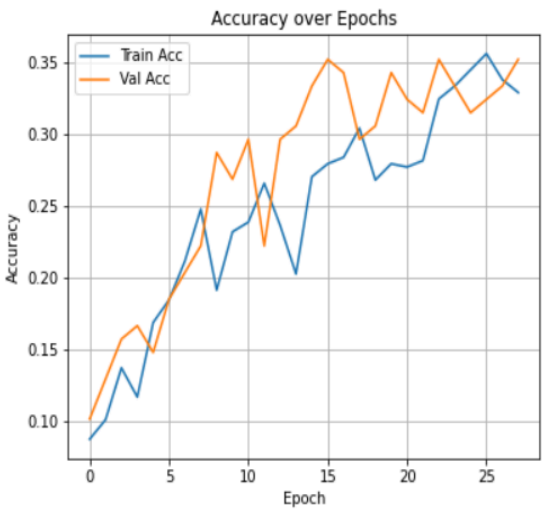
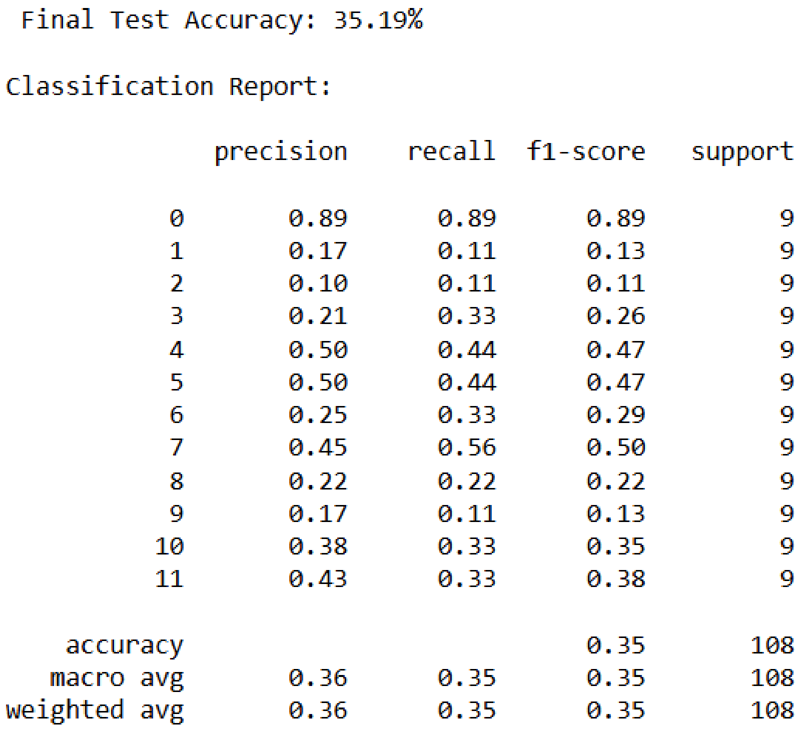


Fig. 6 Accuracy over Epochs (MultiClass)

Fig. 5 and Fig. 6 shows Loss and Accuracy over Epochs from Attempt 4 respectively.

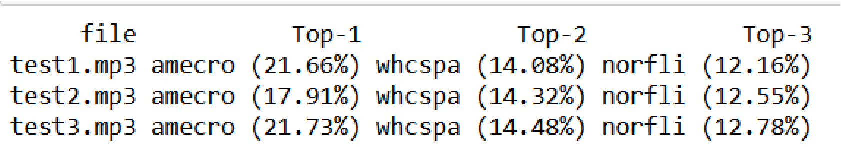
Both the training and validation curve show steady improvement in loss and accuracy throughout training. Validation accuracy peaked at 35% indicating strong generalizations and no signs of overfitting.



```
 Final Test Accuracy: 35.19%

Classification Report:

              precision    recall  f1-score   support

           0       0.89      0.89      0.89         9
           1       0.17      0.11      0.13         9
           2       0.10      0.11      0.11         9
           3       0.21      0.33      0.26         9
           4       0.50      0.44      0.47         9
           5       0.50      0.44      0.47         9
           6       0.25      0.33      0.29         9
           7       0.45      0.56      0.50         9
           8       0.22      0.22      0.22         9
           9       0.17      0.11      0.13         9
          10       0.38      0.33      0.35         9
          11       0.43      0.33      0.38         9

    accuracy                           0.35       108
   macro avg       0.36      0.35      0.35       108
weighted avg       0.36      0.35      0.35       108
```

Fig. 7 Confusion matrix (Multi Class)

The model achieved a final test accuracy of 35.19% across 12 bird species. It performed very well on Class 0, with an F1 score of 0.89 while performance on other classes varied. Classes like 1 and 2 were hard to predict, likely due to overlapping patterns or limited samples. The F1 score of 0.35% shows a fairly balanced effort across all classes.



```
     file           Top-1            Top-2            Top-3
test1.mp3 amecro (21.66%) whcspa (14.08%) norfli (12.16%)
test2.mp3 amecro (17.91%) whcspa (14.32%) norfli (12.55%)
test3.mp3 amecro (21.73%) whcspa (14.48%) norfli (12.78%)
```

Fig. 7 External Test cases

The best model(Attempt 4) predicted all three audio clips with similar results, identifying amecro(American Crow) as the top species (refer to Fig. 7). The second and third most likely predictions were consistently whcspa(White-crowned sparrow) and norfli(Northern flicker). The confidence scores were quite low(all under 22%) which suggests the model was not highly confident and multiple species might be present in each clip. This pattern could indicate overlapping calls or background noise affecting the clarity.

Discussion

One of the main limitations I encountered in this homework was Class Imbalance present in the original dataset. While some bird species had hundreds of samples, others had around 50 or even fewer making it difficult for the model to learn or identify patterns for underrepresented classes.

This was solved through downsampling and class-weighting or focal loss to shift the model focus toward harder-to classify examples. However these solutions introduced trade-offs such as model instability and occasional failure to correctly classify certain species.

Another challenge I faced was the limited size of the dataset even after downsampling. With only 28 samples per class in the final version, the model lacked diversity making it even difficult to classify.
Since the pre-processed shape was already fixed there were less modifications allowed and hence it was difficult to add temporal context (ex. RNN).

Training time was reasonable overall. Binary Classification trained under 1 hour and Multi Class trained around 2 hours. Deeper or regularized models with focal loss or augmentation took more 10-15 mins. Since I used on the fly data augmentation, memory usage remained efficient, but training time increased.

From both Classification report and concussion matrix, it is clear that certain species were significantly harder to predict than others. Class 0 (House Sparrow) was consistently predicted well and achieved high precision and recall. In contrast, Class 1 (Song Sparrow), Class 9 (Spotted towhee) and class 11(White-crowned Sparrow) have very low F1 scores and sometimes close to 0.

After listening to sample audio clips and reviewing the corresponding spectrograms, I observed several patterns among the most challenging classes. These species tend to have shorter, more irregular classes that lack consistent structure.
Additionally, their frequency ranges often overlapped with those of more common species, making it harder for the model to distinguish between them.

The spectrograms for these classes also showed less consistent vertical or horizontal patterns which are typically what CNNs rely on to detect and learn meaningful features.

Furthermore species that were confused may be due to similar structure or unwanted background noise. For instance class 1 and class 9 may be due to overlapping frequency.

While CNNs are a natural choice due to their ability to extract patterns from images other models that could have been explored. A hybrid CNN+RNN model, for example, could be effective where the CNN captures spatial features from the spectrogram and the RNN models the temporal sequence of bird calls.
Another promising direction is the use of pre-trained audio models, which leverage transfer learning and may offer better performance.

Despite these alternatives, CNNs remain a strong choice for bird sound recognition. Spectrograms closely resemble images, making CNNs well suited for capturing spatial features. They allow end-end training eliminating the need for manual feature engineering. Additionally, techniques such as batch normalization, dropout and focal loss allow CNNs to adapt well to challenges like class imbalance and noisy data.

Conclusion

The most striking finding of this report was how much class imbalance and augmentation impacted the models ability to correctly identify bird species. Species with clearer and more distinct vocal patterns were recognized easily , while other overlapping or inconsistent classes led to frequent misclassifications.

The data used for this work consisted of preprocessed, high quality clips from a curated dataset. It would be interesting to rerun these models on continuous or real-time audio, where environmental noise and overlapping species present a greater challenge.

Link to Github

https://github.com/ankitpb/Deep-Learning-for-Bird-Sound-Recognition/tree/main

References

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: with applications in R (2nd ed.). Springer.
2. Combination of RNN and CNN
3. CNN
4. Neural Networks-A beginners guide
5. Layers in CNN
6. Training a CNN from scratch using data augmentation
7. Machine Learning with imbalanced data
8. Focal Loss
9. Neural Networks