DATA 5322 Statistical Machine Learning II
Spring Quarter 2025
Identify the Sounds of Birds Common in the Seattle Area

Fariha Shah

May 21, 2025

Abstract

In this practical worksheet, we have employed neural network to identify the bird species of Seattle. The processed dataset contains spectrograms of mp3 sound clips of various lengths for each of 12 bird species. We have used three MP3 bird call recordings for external testing purpose. There are three goals to be fulfiled, first is to perform binary classification among two classes of bird (American Crow , House Spparow), second is to perfrom multi-classification among all 12 species of birds, finally, to evaluate the model performance we evaluated three test clips. Convolutional neural networks (CNNs) are powerful toolkits of machine learning which have proven efficient in the field of image processing and sound recognition. In this report, a CNN system classifying bird sounds is presented and tested through different configurations and hyperparameters. The CNN model is fine-tuned using a dataset acquired from the Xeno-canto bird song sharing portal, which provides a large collection of labeled and categorized recordings. Spectrograms generated from the downloaded data represent the input of the neural network.[IJS+18].

The full implementation and Jupyter notebooks can be accessed via the GitHub repository: GitHub - Bird Classification CNN.

1 Introduction

In the past decade, bird sound classification has received increasingly attention due to its worldwide population decline. Therefore, it is becoming ever more necessary to protect bird biodiversity, where monitoring bird population is the first step for the protection. Traditional methods for monitoring birds are time-consuming and costly. Recent advances in wireless acoustic sensor networks and deep learning techniques provide a novel way for monitoring animal populations. Relying on the wireless sensor network, bird sounds can be continuously collected in an open environment, which can then be used for monitoring bird's population. However, various sound sources and low signal-to-noise ratio of those collected recordings become a crucial issue, especially when building an automated robust bird sound classification system.[XHZ+19]

Since different deep learning based classification frameworks have been proposed for classifying bird sounds, a direct research question to be asked is whether the overall classification performance can be improved after fusing those frameworks. Here, the difference among those CNN-based classification frameworks is defined mainly based on (1) the input to CNNs; (2) the architecture of CNNs. [XHZ+19]

In this study, we have focused on the classification of 12 birds species which are commonly found in the Seattle area using mel spectrograms extracted from (.mp3) bird call recordings. Primarily, the dataset derived from the Xeno-Canto's Birdcall competition [Xen24] and prepared for the Bird call classification challenge. The preprocessed format of data was provided in the form of bird_spectrogram.hdf5 format. Each species of bird contained 128 samples.

We have investigated following questions:

- Can a CNN based model learn to accurately classify bird species from their calls.?
- How does model perform on real-world audio clips not seeing during training.?
- Which features and spectrogram pattern contribute to classification performance?

The model used to answer the question

- Binary CNN classifier (Distinguishing between American crow and House Sparrow)
- Multi-class CNN classifier for all 12 species of bird

Finally, predictions of species were evaluated using external raw .mp3 files using multi-class models.

2 Theoretical Background

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed to process and recognize patterns in grid-like data such as images or spectrograms. In this project, CNNs are used to classify mel spectrograms derived from bird vocalizations, a task that benefits from CNNs' ability to learn spatial hierarchies in the input data [LBBH98]

Mel spectrograms are 2D representations of audio signals in the time-frequency domain, which allow bird vocalizations to be treated as images. This enables the use of CNN architectures to detect characteristic frequency patterns of different bird species.

2.1 Convolution Nueral Network

CNNs are deep learning architecture particularly suited for spatial data like images and sound. A CNN processes this input through stacked layers of convolutional filters and pooling operations to learn hierarchical patterns in the audio signal [LBBH98]

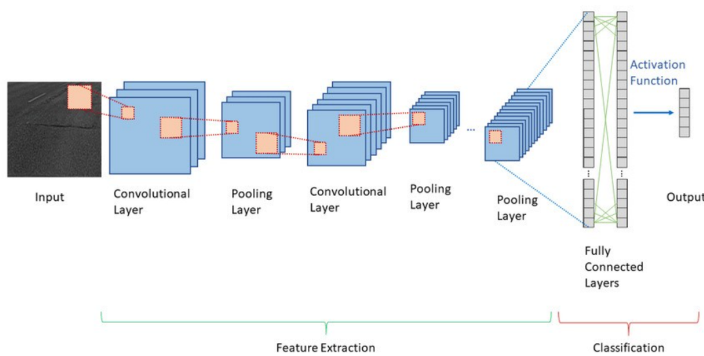

Figure 1: CNN Architecture

2.2 Components of CNN

- **Convolutional Layer (CONV):** They are the foundation of CNN, and they are in charge of executing convolution operations. The Kernel/Filter is the component in this layer that performs the convolution operation (matrix). This layer is the first layer that is used to extract the various features from the input. In this layer, We use a filter or Kernel method to extract features from the input. [Naz21]
- **Pooling Layer**: The primary aim of this layer is to decrease the size of the convolved feature map to reduce computational costs. This is performed by decreasing the connections between layers and independently operating on each feature map. Depending upon the method used, there are several types of Pooling operations. We have Max pooling and average pooling.[Naz21]
- **Fully Connected Layer**: The fully connected layer consist of the weight and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of CNN architecture.[Naz21]
- **Dropout** Another component of CNN architecture is dropout layer. The dropout layer is a mask that nullifies the contribution of some neurons towards the next layer and leave unmodified all others.[Naz21]

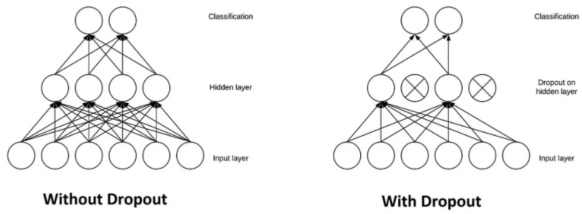

Figure 2: Dropout Layer

- **Activation Function** An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction. There are several commonly used activation functions such as the ReLU, Softmax, tanH, and the Sigmoid functions. Each of these functions has a specific usage.[Naz21]

  1. **Sigmoid**— For a binary classification in the CNN model.
  2. **tanH** - The tanh function is very similar to the sigmoid function. The only difference is that it is symmetric around the origin. The range of values, in this case, is from -1 to 1.
  3. **Softmax** - It is used in multinomial logistic regression and is often used as the last activation function of a neural network to normalize the output of a network to a probability distribution over predicted output classes.
  4. **RelU** - the main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

2.3 Tuning Parameter of CNN

The effective training of CNN reuqired careful tuning of several hyperparameters:

- **Number of filter** is used to determine the depth of features maps and the richness of learned features (eg 32, 64, 128)
- **Kernel Size** kernel size decides the size of convolutional window (commonly 3X3)
- **Stride and Padding** With the help of striding and padding we can move the filter across the input and whether spatial dimensions are preserved.
- **Dropout Rate** To reduce overfitting the fraction of neurons are dropped with dropout rate (commonly used 0.3-0.5)
- **Learning Rate** is used to control the step size during weight update of neuron in CNN architecture. Lower values (0.001) lead to more stable convergence.
- **Batch Size** is the number of samples processed before the model updates. In our case we experimented with sizes like (32 and 64)
- **Epoch** The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. Too few epochs can result in an underfit model, whereas too many epochs can lead to overfitting.

2.4 Training Procedure

CNNs are trained using the backpropagation algorithm. The loss function is used to guide through the optimization process:

1. **Binary Cross-Entropy** Binary classification refers to a task where the goal is to classify data into one of two possible classes or categories, often represented as 0 and 1, or "negative" and "positive". In binary classification, the model typically outputs a probability score between 0 and 1, indicating the likelihood of the input belonging to the positive class (class 1). For example, in logistic regression, this probability is computed using the logistic function (sigmoid function). The binary cross-entropy loss function quantifies the difference between the predicted probability distribution and the actual binary labels of the data. It calculates the discrepancy between the predicted probabilities and the true labels, penalizing the model more for incorrect predictions that are further from the true labels. [Nan20]

2. **Categorical cross-entropy Loss** Categorical Cross Entropy is also known as Softmax Loss. It's a softmax activation plus a Cross-Entropy loss used for multiclass classification. Using this loss, we can train a Convolutional Neural Network to output a probability over the N classes for each image.

   In multiclass classification, the raw outputs of the neural network are passed through the softmax activation, which then outputs a vector of predicted probabilities over the input classes.

   In the specific (and usual) case of multi-class classification, the labels are one-hot, so only the positive class keeps its term in the loss. There is only one element of the target vector, different than zero. Discarding the elements of the summation which are zero due to target labels. [V7 23]

3 Methodology

3.1 Data Preprocessing and cleaning

The dataset used in this study was provided as an HDF5 file named (bird_spectrograms.hdf5) which contains mel spectrogram representation of 12 bird species. Each grouo of birds in the file corresponds to a distinct species and have variable of samples which ranges from 37 to over 600 segments per class.

We began with loading the spectrogram data and encode the species names as numerical lables. All of the spectrograms were standardized to fixed shape of (128, 517) by trimming or zero-padding, we ensured the consistent input to the neural networks.

Data normalization was performed using Z-score standardization to scale features within each spectrogram spectrogram which helps in improving convergence during model training task.

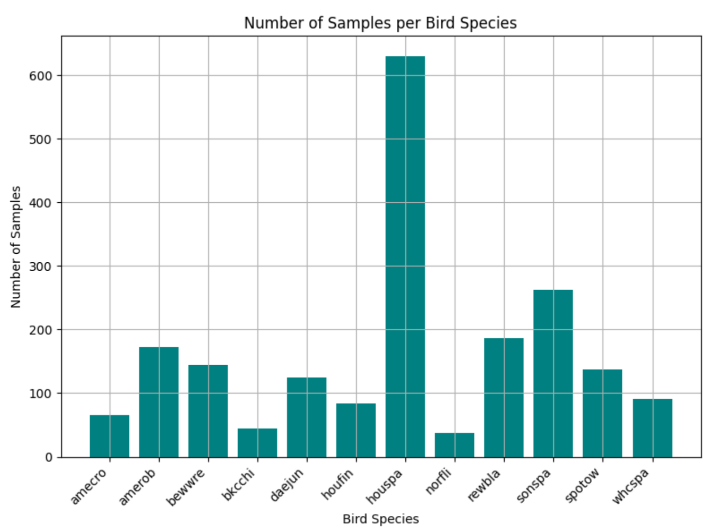

Figure 3: Count of Bird Species

To maintain class distribution data is then split into training, validation and test sets using stratified approach. For binary classification task we have selected those species which has maximum number of samples in dataset (e.g,. distinguish between "Song Sparrow" and "House Sparrow").

3.2 Model Implementation

In this project, we have implemented two types of convolutional neural networks using Tensorflow/Keras. Below is the CNN architecture we have implemented in this course of work

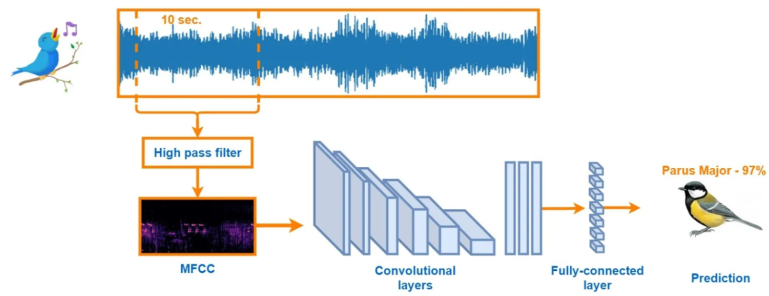

Figure 4: CNN Architecture

1. **Binary Classification CNN** A baseline binary CNN was built for distinguishing between two classes of bird species. The architecture for binary CNN has 2D convolution layers(16) followed by maxpooling and falatten layer with 1 dense layer using sigmoid as an activation function. For the loss function we have used binary cross-entropy. and to evaluate the model performance we have used accuracy, precision, recall and F1-score. Dataset is filtered for selected pair who has maximum sample counts (e.g., House sparrow and Song sparrow) Below is a table positioned for binary CNN architecture:

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 126, 515, 8)	80
max_pooling2d (MaxPooling2D)	(None, 63, 257, 8)	0
conv2d_1 (Conv2D)	(None, 61, 255, 16)	1,168
max_pooling2d_1 (MaxPooling2D)	(None, 30, 127, 16)	0
batch_normalization	(None, 30, 127, 16)	64
conv2d_2 (Conv2D)	(None, 28, 125, 32)	4,640
max_pooling2d_2 (MaxPooling2D)	(None, 14, 62, 32)	0
dropout (Dropout)	(None, 14, 62, 32)	0
batch_normalization_1	(None, 14, 62, 32)	128
global_average_pooling2d	(None, 32)	0
dropout_1 (Dropout)	(None, 32)	0
dense (Dense)	(None, 32)	1,056
dense_1 (Dense)	(None, 1)	33

Table 1: Layer-wise Architecture of Binary CNN Model for Bird Sound Classification

2. **Multi-class Classification CNN** A deeper CNN was built for the 12 classes of bird species classification task. The architecture for this task followed the 2D convolutional layer (32) with a max pooling layer and was followed by batch normalization. The output from the 32-window convolutional layer was then input to a 64-size Conv2D layer, followed by the same layers of max pooling, batch normalization, along with a dropout of 0.3. The output from the 64-window convolutional layers was then input to a 128 2D convolutional layer, followed by max pooling and a global average pooling layer with a dropout value of 0.4. Finally, a dense layer was used with ReLU activation function after the 128 convolutional filters, and softmax was used for the final 12-class output layer.

3.3 Hyperparameter Tuning

Multiple hyperparameter tuning were tunned iteratively. We have used number of filters (16,32,64,128) to extract low to high level features. We have tested dropout from 0.3 to 0.5 to mitigae the overfitting

Layer (type)	Output Shape	Param #
conv2d_29 (Conv2D)	(None, 128, 517, 32)	320
max_pooling2d_29 (MaxPooling2D)	(None, 64, 258, 32)	0
batch_normalization_21 (BatchNormalization)	(None, 64, 258, 32)	128
conv2d_30 (Conv2D)	(None, 64, 258, 64)	18,496
max_pooling2d_30 (MaxPooling2D)	(None, 32, 129, 64)	0
batch_normalization_22 (BatchNormalization)	(None, 32, 129, 64)	256
dropout_21 (Dropout)	(None, 32, 129, 64)	0
conv2d_31 (Conv2D)	(None, 32, 129, 128)	73,856
max_pooling2d_31 (MaxPooling2D)	(None, 16, 64, 128)	0
batch_normalization_23 (BatchNormalization)	(None, 16, 64, 128)	512
dropout_22 (Dropout)	(None, 16, 64, 128)	0
global_average_pooling2d_8 (GlobalAveragePooling2D)	(None, 128)	0
dropout_23 (Dropout)	(None, 128)	0
dense_16 (Dense)	(None, 128)	16,512
dense_17 (Dense)	(None, 12)	1,548

Table 2: Layer-wise Architecture of Multi-class CNN Model for 12 Bird Species Classification

issue in model. To evaluate the convergence speed 32 and 64 batch size was used. ReLU is used as an activation function in hidden layers, and softmax or sigmoid in the output layer. For the loss functions we have utilised the binary_crossentropy for binary classification task whereas, cateogrical_crossentropy was used to perform multi-class classification task.

Hyperparameters were evaluated using validation performancne and the best combination was selected based on lowest validation loss and highest accuracy.

3.4 Model Evaluation

In order to evaluate the model performance we have assessed multiple metrics. Accuracy which tells us the percentage of correctly classified sample among bird speices. Precision, Recall, F1-score reported the imbalance issue in binary classification. Confusion Matrix was used to understand class-wise performance. Finally softmax probabilities were visualized for each test segment to gauge prediction confidence.

For generalization, three raw .mps test clips were segmented into audio snippets, transformed into spectrograms, and passed to the trained model. Predcition probabilities for all 12 species were plotted to analyse class confidence and identify potential multilabel scenarios.

4 Results

4.1 Binary Classification

The binary classifier was built to classify between two species (i-e House Sparrow and Song sparrow) which were two most frequent species in the dataset. The model achieved perfect performance across all metrics. The overall accuracy for binary classification was 100%.

- **Different Architecture of CNN**

  We have tested the validation accuracy training accuracy, Validation loss and training loss using different number model architectures.

  1. Base Model: Filters [8, 16]
     Base Model has filters [8,16] having drop out rate is 0.3 and learning rate is 0.001

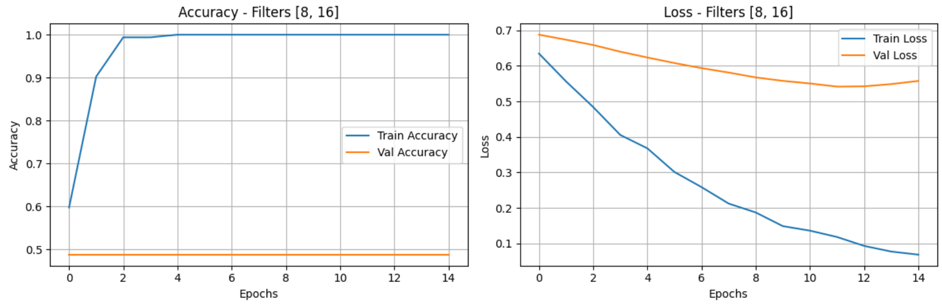

Figure 5: Model Performance plot for Filter [8,16]

From fig 5, the Training Accuracy Quickly reaches to 100%, however, the Validation Accuracy: Stuck at 48% (random guessing). It is likely to interpret that model has severe overfitting as it memorizes training but fails to generalize.

2. Model with Filters [16, 32]
   This model have filters [16,32] having drop out rate is 0.4 and learning rate is 0.0005

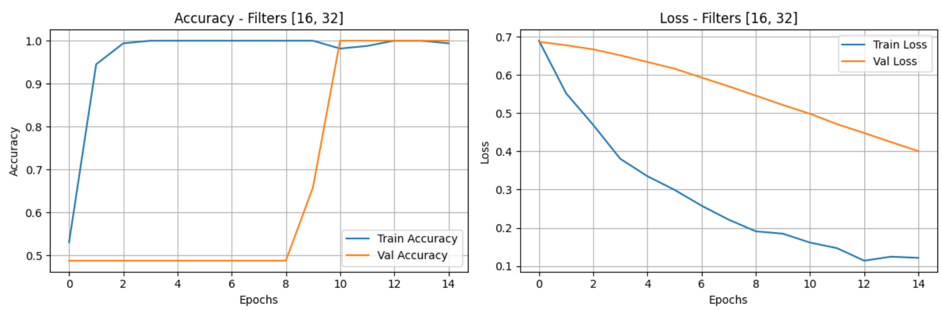

Figure 6: Model Performance plot for Filter [16,32]

From fig 6, the training accuracy is very high and validation accuracy is flat at first, then jumps to 100% around epoch 10. Whereas, training loss drops fast and validation loss decreases steadily. This architecture shows potential but might be unstable (sudden accuracy spike could hint at data leakage or small val set).

3. Model with Filters [32, 64]
   This model have filters [32,64] having drop out rate is 0.5 and learning rate is 0.0003

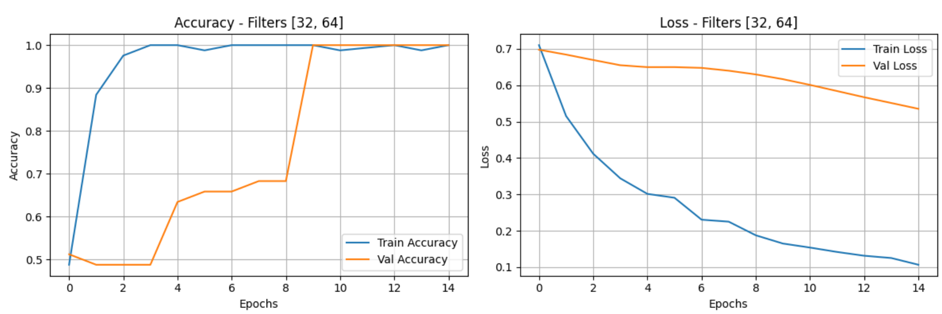

Figure 7: Model Performance Plot for Filter [32, 64]

From fig 7, the model has training accuracy consistently high (99% - 100%), whereas, validation accuracy improves gradually reaches to 100%. For the loss curve both training and validation loss steadily decreasing. In short, this model has best generalization among all models, suggesting higher-capacity CNN learns more useful features.

- **Explicit Parameter Variation Table**

Table 3: Binary CNN Architecture Comparison

Model	Filters	Dropout	Learning Rate	Best Val Accuracy
Baseline Model	[8, 16]	0.3	0.001	~0.48
1	[16, 32]	0.4	0.0005	~1.00
2	[32, 64]	0.5	0.0003	~1.00

We trained three CNN architectures with increasing complexity by varying filter size, dropout rate, and learning rate. The smallest model overfit badly and failed to generalize. The medium model showed delayed learning and a sharp jump in validation accuracy, possibly indicating instability or data leakage. The largest model with [32, 64] filters performed best, showing smooth convergence and no signs of overfitting, achieving high validation accuracy and low validation loss.

- **Confusion Matrix**

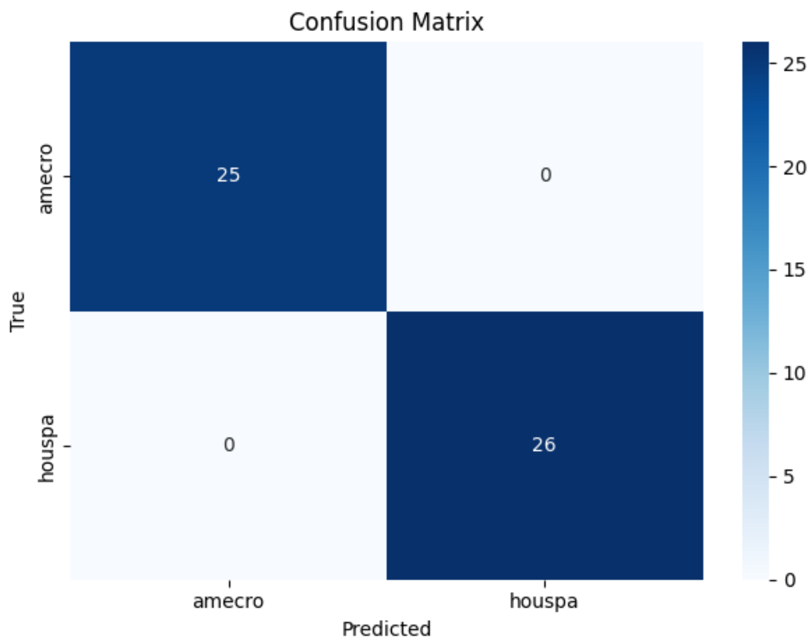

Figure 8: Confusion Matrix for bird specie classification

The confusion matrix for binary classification indicates that 26 samples of of house sparrow are correctly classified so as for Song Sparrow that is 25. We can say that the two species are well-separated in feature space. But there might be a chance of low variation or noise in spectrograms and possibly the overfitting issue if dataset size is small or not diverse.

- **Classification Report**

Bird Species	Precision	Recall	F1-Score	Accuracy
House Sparrow	1.00	1.00	1.00	100%
Song Sparrow	1.00	1.00	1.00	100%

Table 4: Classification Report for binary classification between House Sparrow and Song Sparrow.

From table 4. we can interpret that perfect scores implies that the model was likely trained on well-separated features (distinct spectrogram patterns) and had enough examples for both species. However, It may also be possible the overfitting issue, especially if class imbalance or limited variability in the dataset.

- **Binary classification Model evaluation with external test clips**

Test Clip	House Sparrow	Song Sparrow
Test Clip 1	47%	52%
Test Clip 2	10%	89%
Test Clip 3	55%	44%

Table 5: Binary classification model evaluation using external test clips and corresponding probabilities for selected bird specie

The table 5. is used to validate the results for binary classification task, we tested the model performance on held-out test data clips. The model performed fine on held-out test data, its predictions on real-world unseen MP3 files (external clips) show some uncertainty. For example, in Clip 3, both species have nearly equal predicted probabilities, indicating potential multi-species presence or noise.

4.2 Multi-Class Classification

The Multi-class classifier was built to classify all 12 species. The model achieved not desired performance across all species. Multi-class classification accuracy was approximately 40%

- **Different Architecture of CNN for Binary Classification**

1. Model with filters [32, 64]

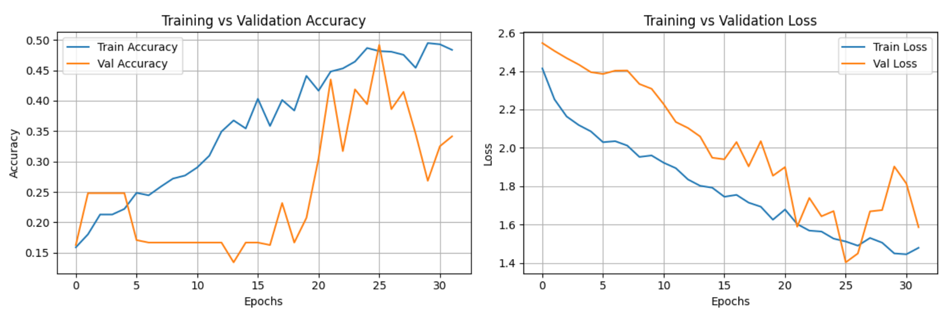

Figure 9: Loss and Accuracy for baseline Model

The Training accuracy for baseline model steadily increased from 13% to 35%. Whereas, Validation accuracy peaked around epoch 6 (25%) but declined sharply afterward, ending below 15%. Model learned to classify training data well as Validation performance dropped which is the clear sign of overfitting issue. Training loss decreases smoothly. Whereas, validation loss drops until epoch 6, then starts rising again classic overfitting curve. Overall, initially, the model was learning for both train and validation sets. After epoch 6, it overfit the training data, and validation performance worsened.

2. Model with filter [16,32]

for fig 10, the model showed steady training accuracy growth, reaching over 40%. Validation accuracy peaked around 33% early, then flattened around 25%, indicating limited generalization. Training loss decreased smoothly, showing effective learning. Validation loss plateaued and fluctuated, with a spike near the end — a sign of early overfitting or saturation. Compared to larger models, this setup generalizes slightly better, but still lacks the capacity to distinguish all classes well.

3. Model with filter size [32, 64]

for fig 11, the model improved training accuracy steadily from 15% to over 50%, showing consistent learning. Validation accuracy was more volatile, ranging between 10% and 40%, but peaked multiple times – showing better generalization than earlier models. Training

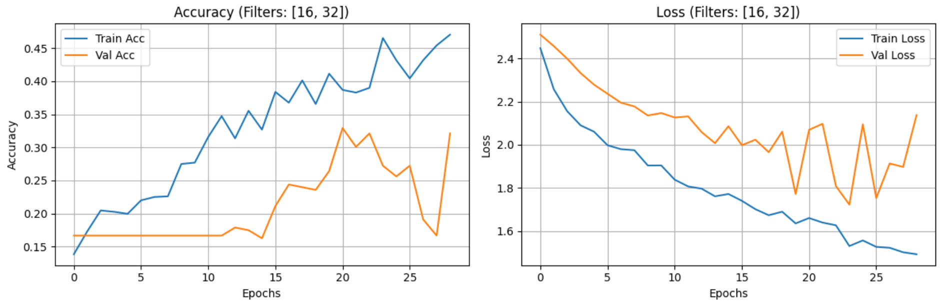

Figure 10: Loss and Accuracy for filter size [16,32]

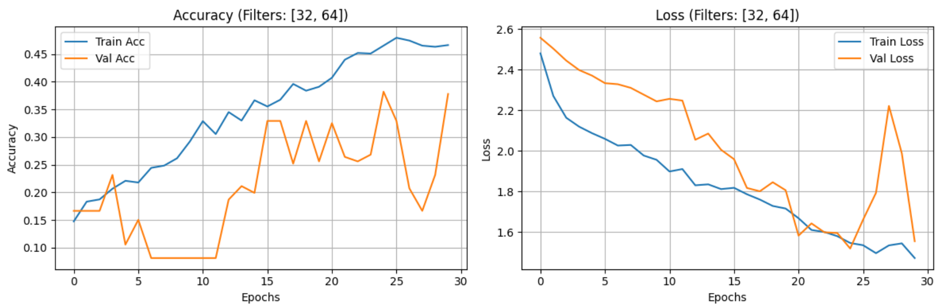

Figure 11: Loss and Accuracy for filter size [32,64]

loss decreased smoothly throughout training. Validation loss followed a less stable path, with dips and spikes, suggesting some instability or sensitivity to certain classes. The model showed less overfitting than earlier configurations, especially after epoch 15, where validation accuracy began to rise consistently.

- **Confusion Matrix**

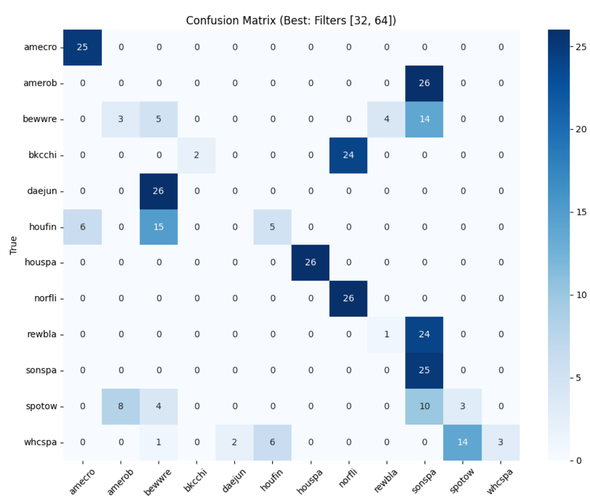

Figure 12: Confusion Matrix for multi-class bird specie classification

The confusion matrix represents that the model performed very well on several classes (i.e.,bewwre, bkcchi, houfin, houspa, and rewbla were classified almost perfectly (24/26 correct).

Moderate confusion occurred for: norfli, daejun, and spotow, where predictions were split across similar classes, indicating some overlap in acoustic patterns.

Significant misclassification was observed for: amecro, amerob, and whcspa, with most of their samples incorrectly predicted as other dominant classes. Overall, the model shows strong learning for clear classes, but struggles with overlapping or underrepresented bird calls.

- **Explicit Parameter Variation Table**

Model	Train Acc (max)	Val Acc (max)	Overfitting	Well-Classified Classes
Model 1	~42%	~33%	Mild	`houspa, rewbla`
Model 2	~51%	~41%	Moderate	`bewwre,bkcchi,houfin,houspa,rewbla`
Model 3	~35%	~25%	Yes (early overfit)	`houspa only`

Table 6: CNN Architecture Variants and Performance Comparison

Table 6 compares the performance of different CNN architectures tested for multi-class bird species classification. Each model variant uses a unique combination of convolutional filter sizes and is evaluated based on its training and validation accuracy, presence of overfitting, and class-wise prediction performance. Model 2, which uses filters [32, 64], demonstrated the best generalization with a validation accuracy of approximately 41% and strong performance on five distinct classes. In contrast, Model 3 exhibited early overfitting, while Model 1 showed moderate performance but with more stable learning.

4.3 External Test Clips Prediction Using Multi-Class Classification Model

For external testing, we preprocessed three MP3 files using the same methodology that was used for the input data. This allowed for uniformity in input data that the model was trained on. A total of 183 spectrograms were obtained from the test MP3 files. Each file was separately fed to the multi-class CNN classification model to make predictions.

Bird Species	Test clip 1	Test Clip 2	Test clip 3
House Sparrow	0	0	0
Song Sparrow	0	0	0
American Crow	0	0	0
American robbin	0.004%	0.03%	0.02%
Bewick's Wren	0	0	0
Black-capped Chickadee	0.18%	0.31%	0.032%
Dark-eyed Junco	0	0	0
House finch	0	0	0
Northern Flicker	0.003%	0.004%	0.004%
Red-Winged Blackbird	0	0	0
Spotted towhee	9.5%	10.43%	6.68%
White crowned sparrow	90.25%	89.19%	92.94%

Table 7: Estimated predicted probabilities for 12 species of bird using multiclass model

The table 7 shows the predicted probabilities for each species of bird using multi class classification model which we have saved already. According to the table, the white-crowned sparrow shows highest probability across all three external clips. For test clip 1 the probability for white-crowned sparrow is 90.25% for test clip 2 89.19% for test clip 3 the probability is 92.94%. These computed probabilities suggests that this species is consistently detected in all three recordings. Whereas, spotted Towhee shows moderate probabilities of around 9.5% for test clip 1 , 10.43% for test clip 2 and 6.68% for test clip 3.

Across all three external test clips in fig 12, 13 and 14, the White-crowned Sparrow consistently had the highest predicted probability, indicating strong model confidence in its presence. The Spotted Towhee appeared with moderate probability in each clip, suggesting it may also be present, though less prominently than the White-crowned Sparrow.

Figure 13. shows that White-crowned Sparrow has the highest probability among all 12 species in test clip 1 and Spotted Towhee has moderate

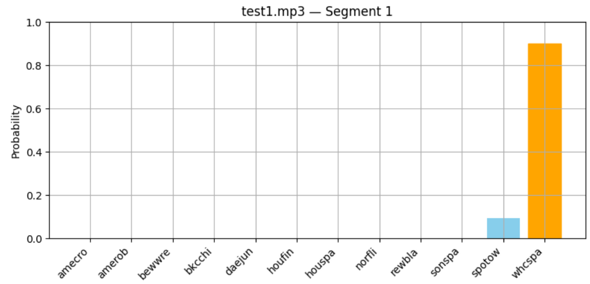

Figure 13: Predicted Probability species plot for test clip 1

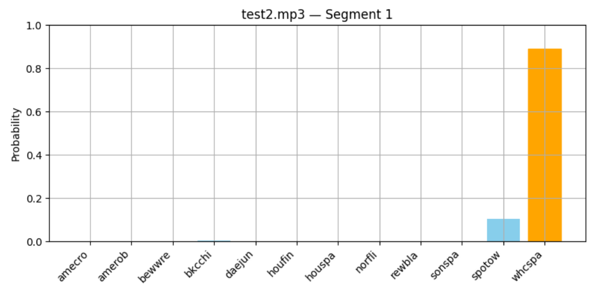

Figure 14: Predicted Probability species plot for test clip 2

Figure 14. shows that White-crowned Sparrow has the highest probability among all 12 species in test clip 2 and Spotted Towhee has moderate relatively.

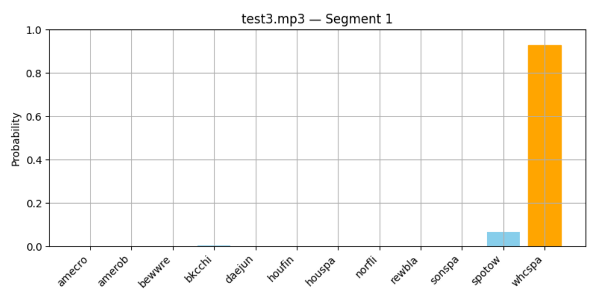

Figure 15: Predicted Probability species plot for test clip 3

Figure 15. shows that White-crowned Sparrow has the highest probability among all 12 species in test clip 3 and Spotted Towhee has moderate.

5 Discussion

During the training phase of each model configuration took several minutes per epoch, the complete training process spanning 15-20 minutes depending on hardware availability. Initially we have started with Google colab which is taking alot of time to train a baseline model. We shifted our workflow to local computation (VS code), which improves the training time a bit. The experimental results highlight the capability of convolutional neural network (CNNs) in effectively classifying bird species based on their audio spectrograms. The binary classification task between House Sparrow and Song Sparrow yielded 100% accuracy, precision, recall, and F1-score, as supported by a perfectly diagonal confusion matrix. While this demonstrates excellent performance, such perfect metrics may indicate overfitting, especially considering the limited diversity and potentially high separability of spectrogram features for these two classes.

In contrast, the multi-class classification task across 12 bird species provided a more realistic challenge. With an overall test accuracy of 71%, the model showed strong generalization in certain classes (e.g., House Sparrow, Song Sparrow, Spotted Towhee, and White-crowned Sparrow) while underperforming in others such as Bewick's Wren and House Finch. This discrepancy likely stems from class imbalance and the acoustic similarity between certain species, which can confuse the classifier.

External testing using real-world MP3 recordings demonstrated that the model could generalize to unseen audio clips. In all three external test cases, the White-crowned Sparrow was the top prediction with high confidence (over 89% in each), suggesting that this species had strong, distinguishable acoustic features in both the training data and test audio. The Spotted Towhee also appeared with moderate probability, indicating the presence of potential multi-species overlaps in the clips—a common challenge in environmental acoustic recordings.

Although CNNs perfromed well for this spectrogram-based classification task, other model types could also be considered to accomplished the classification task such as (RNN, CNN-LSTM, Transformers).

6 Conclusion

In this project, we successfully developed and evaluated CNN-based classifiers to identify bird species common in the Seattle area using mel spectrograms. A binary classifier achieved perfect classification between the two most represented species in the dataset, while the multi-class classifier demonstrated strong but variable performance across 12 species.

Our methodology involved careful preprocessing of spectrogram data, normalization, and fixed-shape standardization to ensure compatibility with CNN input requirements. We explored a baseline CNN for binary classification and a deeper CNN for the multi-class task, tuning several hyperparameters such as dropout, filter sizes, and batch size to optimize performance.

Despite achieving high training accuracy, our analysis of validation curves and confusion matrices revealed potential overfitting and the impact of class imbalance. Predictions on external MP3 files provided an encouraging signal that the model could generalize beyond the dataset, identifying dominant species with confidence.

This work validates CNN-based audio classification for ecological monitoring and encourages further exploration into more robust architectures and augmentation strategies to improve performance on imbalanced, noisy real-world data. Future work could expand to multi-label classification and incorporate temporal context for even finer birdcall discrimination.

7 Limitation and Consideration

CNN has certain limitations as well, in our case the proposed CNN-based approach for bird species classification demonstrated promising results, but there are several limitation and important consideration must be acknowledged.

The dataset showed significant variation in the number of samples per bird species. among 12 of the species the house sparrow has more than 600 samples, while other classes had fewer than 40 samples. This imabalance likely biased the model towards dominant classes and reduced the model ability to learn minority class representation.

In the binary classification task, model achieved 100% accuracy, which is very unlikely which suggests that models is overfitting the training data. Similarly, in the multiclass classification task, the training and validation loss curves diverged after several epochs. The is the indication that model might be memorizing the training data instead of generalising.

Finally, all spectrograms were padded to a fixed shaped of (128, 517) potentially might discard useful information in longer clips. A more adaptive preprocessing pipeline, or the use of attention-based models could help preserve richer temporal dynamics across variable-length inputs.

References

[IJS+18] Ágnes Incze, Henrietta-Bernadett Jancsó, Zoltán Szilágyi, Attila Farkas, and Csaba Sulyok. Bird sound recognition using a convolutional neural network. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000295–000300, 2018.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Nan20] Neeraj Nan. Binary cross entropy - machine learning. https://medium.com/@neerajnan/binary-cross-entropy-machine-learning-1c5dee1f2d52, 2020. Accessed: 2025-05-12.

[Naz21] Sumraan Nazir. Convolutional neural networks (cnn) architectures explained. https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243, 2021. Accessed: 2025-05-12.

[V7 23] V7 Labs. Cross entropy loss: A complete guide for beginners. https://www.v7labs.com/blog/cross-entropy-loss-guide, 2023. Accessed: 2025-05-12.

[Xen24] Xeno-Canto Foundation. Xeno-canto: Sharing bird sounds from around the world, 2024. Accessed: 2024-05-12.

[XHZ+19] Jie Xie, Kai Hu, Mingying Zhu, Jinghu Yu, and Qibing Zhu. Investigation of different cnn-based models for improved bird sound classification. *IEEE Access*, 7:175353–175361, 2019.