Bird Classification based on their song audio

Ishan Narain Rohatgi
Department of Computer Science
North Carolina State University
irohatg@ncsu.edu

Jasmine Madonna Sabarimuthu
Department of Computer Science
North Carolina State University
jsabari@ncsu.edu

Lauren Musa
Department of Mathematics
North Carolina State University
lrmusa@ncsu.edu

1 Background & Introduction

1.1 Problem Statement

It has become important for wildlife monitoring tasks to be able to identify birds based on their unique sounds and calls. Hence, we propose to build machine learning models that correctly identify various bird species from their audio sounds. We specifically are looking into the species: the Greater Yellowlegs, the Arctic Tern, the Verdin, and the Blue Mockingbird. The bird song audio is taken in the wild at various time lengths.

Audio information retrieval is an emerging research area that receives growing attention from both the research community and music industry. One of the direct implications of this research is our topic of Bird Song Classification. Hence, we emphasize on recent development of the techniques and discusses several open issues for future research.

1.2 Literature Survey

Audio classification in general has been widely studied, with applications to human speech and music being the most common. The work by Fagerlund [1] was one the earliest works to classify birds based on their sound using SVM. He classified 2 sets of bird species and represented the bird sounds as two different parameters: (i) the mel-cepstrum parameters and (ii) a set of low-level signal parameters. Additionally, the more recent work by Kahl, et al.[3] uses CNNs which perform better for automated processing of field recordings. They generate deep features based on visual representations of audio recordings. This has proven to be very effective when applied to the classification of audio events.

2 Method

Audio signals vary from each other and contain a lot of information based on their frequency distribution, decibel level changes, spectral contrast etc. Hence, we must first understand to extract these features from audio files and discern different features from each other. We propose to parse individual audio files and convert each audio file into STFT (Fourier transformed) images to extract information from it. We will build a simpler model first using low level features as a baseline.

The first model is developed using a Support Vector Machine (SVM), a discriminative classifier, which identifies a hyperplane in an N-dimensional space that distinctly classifies data points into categories. SVMs have been used in the past for phonetic segmentation, speech recognition, and general audio classification. One advantage of SVMs is their accuracy and superior generalization properties they offer when compared to many other types of statistical classifiers. SVMs are based on statistical learning theory and structural risk minimization.

We also propose to use Convolution NeuralNetworks (CNN) which are known for extracting features from images and predicting suitable outcomes. The last model will be a recursive CNN model that takes buckets of audio sequences and maintains context while training. A comparative analysis of each model will be made.

3 Plan and Experiment

3.1 Dataset

Our data was retrieved from an online bird song database hosted on **(https://www.xeno-canto.org/)** which is run by a foundation in the Netherlands. The database receives bird song recordings from users all around the world with basic information regarding the collection of the data. The identification of the bird songs are vetted by xeno-canto administration and users alike. If an identification of a bird song is in question, the audio file is taken out of the database. The database does not accept recordings of captive birds. We created a python script using xeno-canto's official API documentation to download audio files (.mp3) and metadata (species, country found, scientific name etc) of 600 recordings. Only 4 species of birds will be analyzed. The species were selected based on the availability of the data; each species had roughly 150 recorded songs.

3.2 Hypothesis and Goal

We believe that the CNN architectures would be better models for classification than the SVM. The SVM only uses low level audio features; whereas, the spectogram images that are fed into the neural networks mimic mid level audio features. Mid level audio features such as pitch, rhythm, and harmony are what humans use in classifying bird songs.

Additionally, between the CNN models, the recursive CNN should perform with greater accuracy. The CNN models may not preserve the temporal relationship in the sound signals. Hence, adding a parallel RNN layer would theoretically, maintain context while making predictions.

3.3 Experimental Design

The audio files were processed using python's LibROSA audio analysis package. This package makes use of the continuous audio files by sampling the data (xVolts) into a vector which is then normalized. This provided a numerical representation of the data. From there, we preformed transformations for feature extraction.

The features we pulled from the audio segments are low level features that describe the quality of the sound. The extraction of these features are well documented and easy to implement, and , as Fu et. al has said, "... [They] have demonstrated good performance in virtually all music classification tasks."[2] As such, we believe they can preform well in classifying wild bird songs.

These low level features include the root-mean-square energy (RMSE),the spectral centroid, the spectral bandwidth, the roll-off frequency, the zero-crossing rate, and the Mel-frequency cepstral coefficients, as which there will be 20. The spectral centroid finds the center of mass of the spectrum.

The spectral bandwidth is the measure of the width of the half maximum peak of a signal. The roll-off frequency acts as a filter point. It is used to filter out harmonics of the wave form that fall below or above the center to a set percent. The roll-off captures 85% of the energy. Since we are using bird song recorded in the wild, the roll-off can act as a filter for white-noise. The zero-crossing rate measures the sign change rate of the data. Lastly, the mel-frequency cepstral is a non-linear transformation of the spectral data that highlights period components of the signal. (Fu et. al).

3.3.1 Dimensionality Reduction

The bird song data set was split into training and testing data at a 4 to 1 ratio using stratified sampling. The training and testing data set was normalized using the max and min values of each feature. A principal component analysis (PCA) was preformed on both the training and testing set. The number of principal parts were determined by the eigenvalue and eigenvector decomposition of the covaraince matrix.

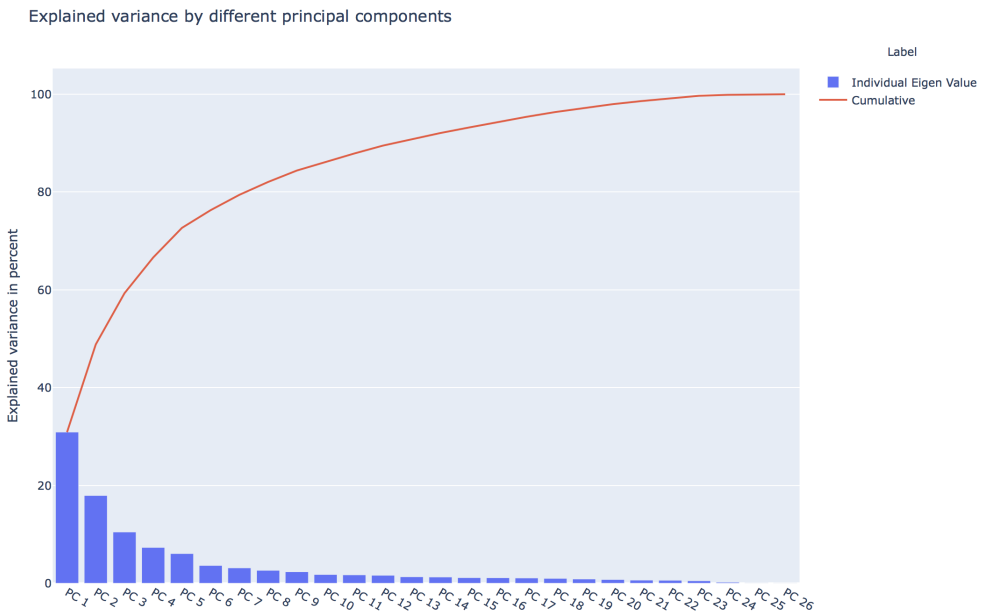
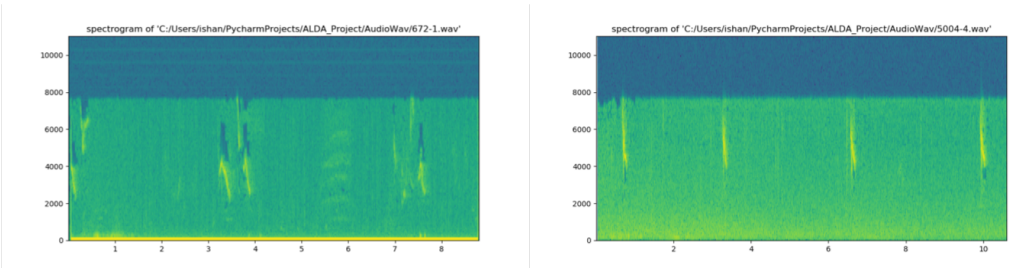


Figure 1: eigenvalues of the normalized training set's covariance matrix vs the explained variance.

While preforming the PCA to feed the SVM, we found it required 10 principal components to explain 85% of the variance. Figure 2 shows the eigenvalues of the normalized training data.

3.3.2 Spectrogram Generation For Neural Networks

A Short Time Fourier Transform (STFT) was applied to each audio file. The data was broken into over lapping time segments which were then transformed by a Discreet Fourier Transform (DFT). The DFT provided sinusoidal decomposition of the data. Thus we learned how the frequency and amplitude of the bird song changes with time. The information is displayed in a spectrogram as seen in Figure 2. The other features are derived from performing calculations on each segment. These features, STFT and resulting spectrograms, are dependent on a selected sampling rate of 22050 Hz.



(a) Blue Mockingbird (b) Verdin

Figure 2: Example Spectrogram for Bird Species

4 Results

4.1 Support Vector Machines

The PCA transformed training data is used to fit the Support Vector Machine (SVM) model. The PCA transformed testing data is used to predict the bird species. The svm model is initialized with the following parameters : C=1, kernel=rbf and gamma=10. The results obtained from PCA decomposition allowed us to determine the number of features to capture maximum variance while reducing dimensionality.

Using the ideal number of PCA components , we utilised sklearn's GridSearch cross-validation library to tune the SVM model hyper-parameters. We created a grid of hyper-parameters and just try all of their combinations till we get a model with optimal performance.

We mapped the bird species using label encoding as follows: 0 - Arctic tern, 1 - Blue Mockingbird, 2 - Greater Yellowlegs, 3 - Verdin. After splitting the data set into training and testing, we got the following division of class labels.

Class	Training	Testing
0	121	30
1	117	33
2	102	47
3	110	40
Total	450	150

Table 1: Number of records per class label in training data

We iterated through the following values (2, 4, 8, 10, 15, 20) for number of Principal components and predicted the bird species with the testing data. We found that our SVM Model attains a maximum accuracy of 68.667% when we selected 10 principal components. The Figure 3 shows the accuracy of our model with changing number of principal components. Hence, we say that 10 is the number of principal components that capture maximum variance in the data and these 10 features should be utilised for training and tuning the model.

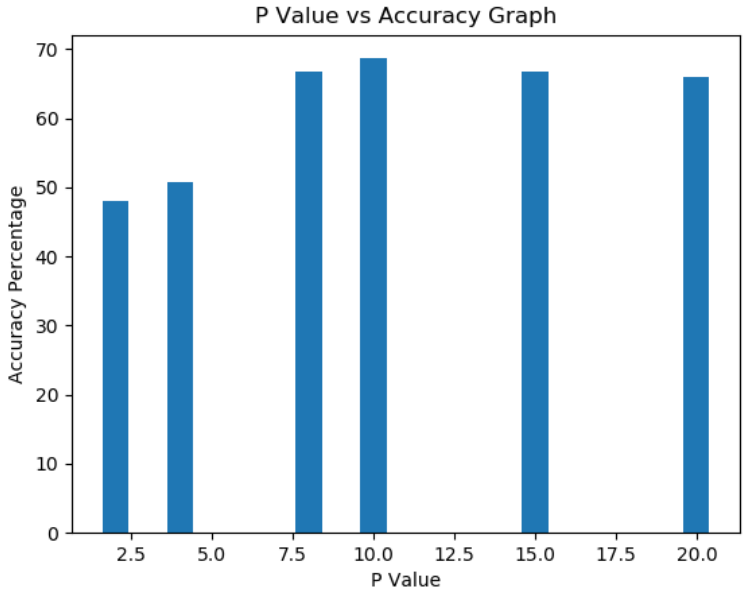


Figure 3: Accuracy of SVM model for changing number of principal components (P)

For hyper-parameter tuning the following grid search parameters were chosen -:
Penalty 'C': [0.1, 0.5, 1, 5, 10, 20, 100, 1000]
Influence 'gamma': [10, 5, 1, 0.1, 0.01, 0.001, 0.0001]
Transformation 'kernel': ['rbf', 'sigmoid', 'poly']

The best results were found for the following parameters-:
'C': 5, 'gamma': 10, 'kernel': 'rbf'

and our final model after hyper-parameter tuning, looks as follows-:

Class Label	Precision	Recall	F1-Score	Support
0	0.69	0.83	0.76	30
1	0.59	0.61	0.60	33
2	0.74	0.66	0.70	47
3	0.74	0.70	0.72	40
accuracy			0.69	150
macro avg	0.69	0.70	0.69	150
weighted avg	0.70	0.69	0.69	150

Table 2: Classification Report

SVC(C=5, cache_size=200, class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3, gamma=10, kernel='rbf', max_iter=-1, probability=False, random_state=None, shrinking=True, tol=0.001, verbose=False)

After running, the test data on the above tuned model, we can see the classification report as shown in Table 2. As can be seen, we get an accuracy of 69% for the best parameter for SVM model.

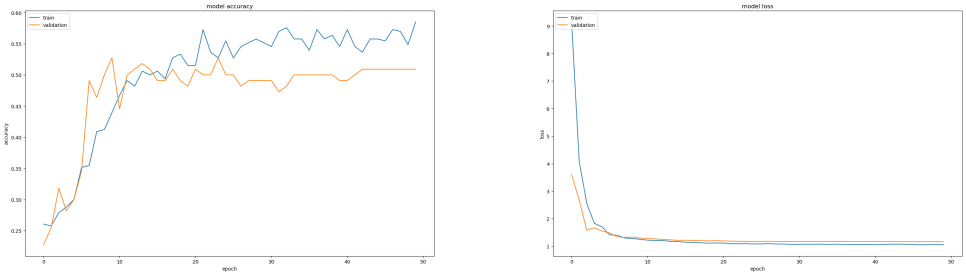
4.2 Convolutional Neural Network

The raw audio signals from which the spectrogram was created, were fed into the CNN (because of memory error problems with the images). After solving these errors we also fed the spectrogram images into the CNN, keeping the resolution of images as 256x256.

The CNN architecture, contained of 3 convolution layers with ReLu activation function. We also added MaxPooling layer after each Convolution layer for discretization and to reduce dimensionality, and a Dropout layer to reduce over-fitting the model. The final layer was a Dense layer with Softmax activation to allow us to classify the Bird Species into one of 4 classes.

The data was split into a ratio of 3:1:1 for training, validation and testing respectively.

For hyper-parameter tuning the following grid search parameters were chosen -:
Optimizer : ['adam', 'Adagrad', 'RMSprop']
Batch Size : [16, 32, 64, 128, 256]
Epochs: ['25', '50', '100']



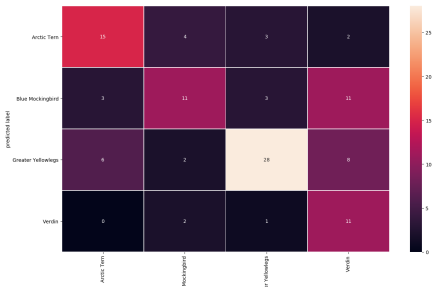
(a) Accuracy of CNN model

(b) Loss for CNN

Figure 4: CNN results using audio signals

The best results were found for the following parameters-:
'Optimiser': Adam, 'Batch Size': 64, 'Epochs': 50

Additionally, checks were setup to Monitor Validation Accuracy while training. Model



(a) Confusion Matrix

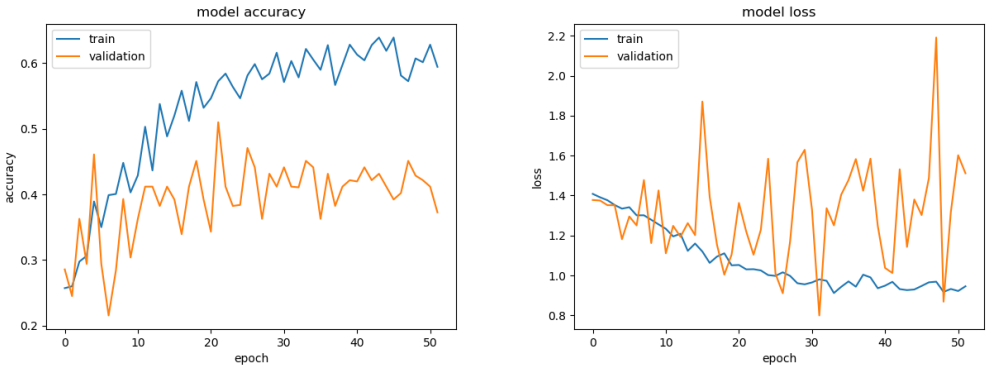
	precision	recall	f1-score	support
Arctic Tern	0.62	0.62	0.62	24
Blue Mockingbird	0.39	0.58	0.47	19
Greater Yellowlegs	0.64	0.80	0.71	35
Verdin	0.79	0.34	0.48	32
accuracy			0.59	110
macro avg	0.61	0.59	0.57	110
weighted avg	0.64	0.59	0.58	110

0.5909090909090909

(b) Classification report

Figure 5: CNN results using audio signals

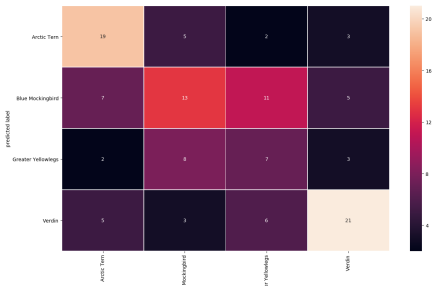
Checkpoint was used to save the weights for the model which yielded the best validation accuracy. We additionally, wanted to reduce the learning rate for the optimizer if validation accuracy remained constant for 5 epochs. We also wanted to terminate the training earlier if validation accuracy did not improve for 20 epochs.



(a) Accuracy of CNN model

(b) Loss for CNN

Figure 6: CNN results using Spectrogram Images



(a) Confusion Matrix

	precision	recall	f1-score	support
Arctic Tern	0.66	0.58	0.61	33
Blue Mockingbird	0.36	0.45	0.40	29
Greater Yellowlegs	0.35	0.27	0.30	26
Verdin	0.60	0.66	0.63	32
accuracy			0.50	120
macro avg	0.49	0.49	0.49	120
weighted avg	0.50	0.50	0.50	120

0.5

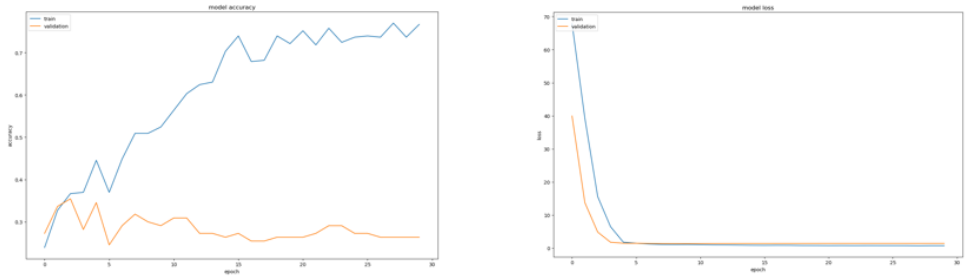
(b) Classification report

Figure 7: CNN results using Spectrogram Images

From the results above, we achieved a maximum accuracy of 60%, lower than our SVM model. These results were for the CNN model which was fed the audio signal, and not the spectrogram images.

4.3 Recurrent Convolutional Neural Network

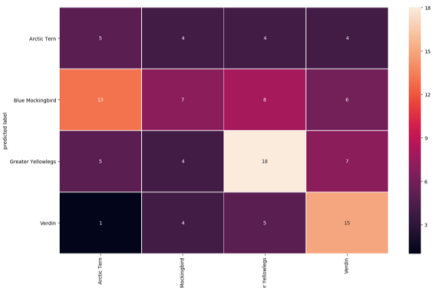
RNNs excel in understanding sequential data by making the hidden state at time t dependent on hidden state at time t-1. The spectograms have a time component and RNNs can do a much better job of identifying the short term and longer term temporal features in the song. Hence, the last model we propose will be a recursive CNN model that takes buckets of audio sequences and maintains context while training and should theoretically better classify various transformed images.



(a) Accuracy of Recursive CNN model

(b) Loss for Recursive CNN

Figure 8: Recursive CNN results using audio signals



(a) Confusion Matrix

	precision	recall	f1-score	support
Arctic Tern	0.29	0.21	0.24	24
Blue Mockingbird	0.21	0.37	0.26	19
Greater Yellowlegs	0.53	0.51	0.52	35
Verdin	0.60	0.47	0.53	32
accuracy			0.41	110
macro avg	0.41	0.39	0.39	110
weighted avg	0.44	0.41	0.42	110

0.4090909090909091

(b) Classification report

Figure 9: Recursive CNN results using audio signals

We added a parallel LSTM layer, with the CNN architecture proposed above. The idea behind this was that the temporal relationships may not be preserved by the CNN. The final output layer in this case was a concatenation of the CNN and RNN blocks. Final output layer is a Dense layer with softmax activation.

For hyper-parameter tuning we followed the same approach as we did for the CNN. The best results were found for the following parameters-:
'Optimiser': Adam, 'Batch Size': 64, 'Epochs': 50

From the results above, we achieved a maximum accuracy of 41%, lower than our CNN and SVM models. The model easily overfits and does not converge well. From above we can see that the CNN model did not perform massively better using spectrogram images. Hence, the CNN-RNN is also unlikely to give better results, reasons for which have been explained later.

4.4 Evaluation of Results

Are bird sounds complex (different) enough? Compared to human voice and musical songs, bird sounds operate on a much lower level of frequency. Hence, this low difference in frequency level change may trick more complex models while classifying bird species. Maybe, for this reason we should still rely on fitting low-level audio features using statistical models to best classify bird sounds.

We expected CNN and CNN-RNN to perform better than SVM, but contrary to our expectation we got poor results for CNN models. Previous works [3] have shown acceptable results with CNN based models too. But, they extracted many 4 second spectrograms from the long audio and removed spectrograms with background noise. These spectrograms were used as input for the CNN model. We believe using spectrogram images after doing the necessary pre-proessing on the spectrograms similar to Kahl et al.[3] will provide better results for the Neural Network models as well. We pursue this as our future work.

5 Conclusion

In this project, we classified 4 different bird species based on their sound audio. We used three machine learning models for the classification - Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Recursive CNN respectively. Based on our experiments, we got the best results for the SVM model with an accuracy of 69%. The CNN and recursive CNN models performed poorly when compared to SVM. The CNN model and recursive CNN models gave highest accuracy of 59% and 41% respectively.

From the models presented in the preceding sections, it is clear that the challenge of bird species classification based on bird song audio is a suitable candidate for deep learning exploration. With the proposed models, it is guaranteed to achieve satisfactory results as presented in previous sections.

To enable higher performance, some preferred enhancements would be as follows-:

1. Increased training using more data samples (papers read utilised at-least 2-3 thousand recordings).
2. Employ moving window with overlapping stride as a technique for data augmentation.
3. Instead of using the spectrogram image of the long audio (like 4 min length), it is a better idea to use shorter spectrogram images (like a few seconds). Filter spectrogram images to represent audio signals free of background noise.

We have also made our work available to view on our Github Repository here -: https://github.ncsu.edu/irohatg/ALDA_Project_BirdSong

6 References

References

[1] Seppo Fagerlund. Bird species recognition using support vector machines. *EURASIP Journal on Applied Signal Processing*, 2007(1):64–64, 2007.

[2] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010.

[3] Stefan Kahl, Thomas Wilhelm-Stein, Hussein Hussein, Holger Klinck, Danny Kowerko, Marc Ritter, and Maximilian Eibl. Large-scale bird sound classification using convolutional neural networks. In *CLEF (Working Notes)*, 2017.