Rainforest Audio Identification
Craig Fox, Phil Genovese, Jared Huzar, Ben Rapp

Data Report

General Description

[Rainforest Connection Species Audio Detection | Kaggle]

One of the major issues that we face as a planet is the loss of biodiversity in our rainforests. There are many rare species that require careful supervision in the rainforest, and one of the best ways to monitor species health is using audio data. Since some species are hard to find visually, Rainforest Connection has created a product called RFCx, which automatically monitors audio on the forest floor to aid conservation management efforts. Since the presence of rainforest species is an indicator of climate change and habitat loss, being able to monitor rainforest species constantly is key to preserving rainforest health. To train a model, a collection of audio files with minimal labeling along with some metadata is provided.

The audio data is FLAC files. The associated information is divided into two csv tables: one for true positives and one for false positives. The true positives were developed by experts identifying some species throughout the audio clip. The data collectors then developed a simple CNN prediction model, and ran the data through it. The experts then reviewed the predictions and identified cases where it was incorrect. These incorrect marks make up the false positive dataset, i.e. cases where a species is identified as not present in a clip. The interesting part of this data is that it is very incomplete. While we do not know the exact number of species present in a recording, on average only one species is identified at one point in the recording. From qualitative experience listening to the clips, there are at least a hundred calls in each recording. This presents a huge challenge. The other interesting fact is the correlation between the true positives and the false positives. Even though it was a simple model, the false positives are still cases that would be difficult to distinguish. This underscores the importance of the false positive dataset. Incorporating it into the model means it will especially help the cases where our true positive model was most off.

Species Data

During the competition, it was revealed that all the species are either birds or frogs. They are also all present in the same rainforest environment. While other species like insects may be present, these do not need to be identified.

After the completion of the competition the species names were all released. We decided to operate with only the information given during the competition. Therefore, we made our assumptions and models knowing that it was bird and frog rainforest species but nothing else.

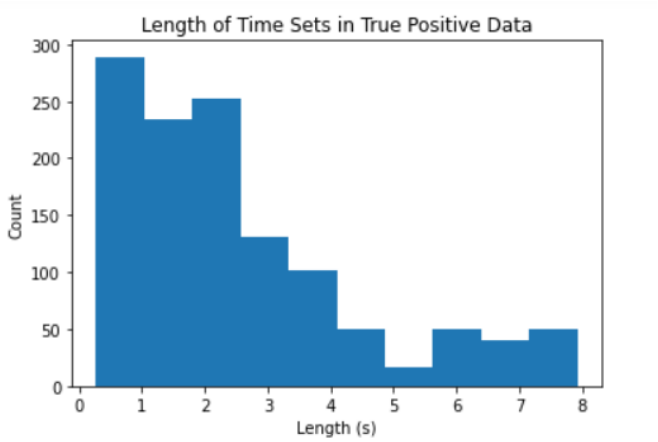
ID Scientific Name
0 Eleutherodactylus gryllus
1 Eleutherodactylus brittoni
2 leptodactylus albilabris
3 Eleutherodactylus coqui
4 Eleutherodactylus hedricki
5 Dendroica angelae
6 Melanerpes portoricensis
7 Coereba flaveola
8 Eleutherodactylus locustus
9 Margarops fuscatus
10 Loxigilla portoricensis
11 Vireo altiloquus
12 Eleutherodactylus portoricensis
13 Megascops nudipes
14 Eleutherodactylus richmondi
15 Patagioenas squamosa
16 Eleutherodactylus antillensis
17 Turdus plumbeus
18 Eleutherodactylus unicolor
19 Coccyzus vieilloti
20 Todus mexicanus
21 Eleutherodactylus wightmanae
22 Nesospingus speculiferus
23 Spindalis portoricensis

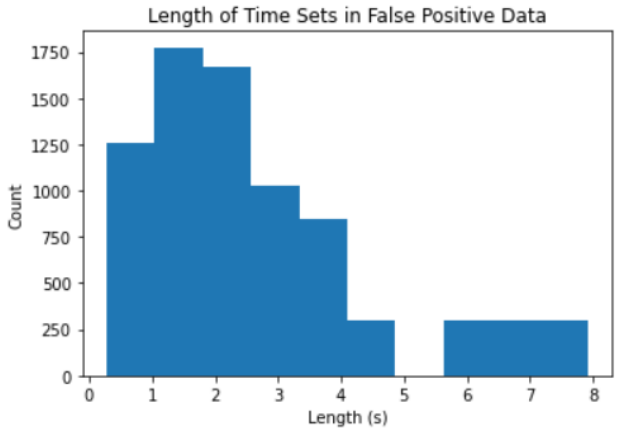
Metadata Analysis

The first piece of crucial information is the comparative size of the true and false positives. There are 1216 true positive identifications and 7781 false positive

identifications. This means the false positives will also be crucial because of how much more data they provide. Each identification contains seven pieces of information: recording_id, species_id, songtype_id, t_min, f_min, t_max, and f_max. The recording_id represents the audio clip used. The species_id is the integer representing which of the 24 species is being identified. Songtype_id is the integer that represents which song of that species is identified. Only 1 and 4 are used, with 1 being the primary call and 4 being a secondary call. Every species has 1 but only 2 species (17 and 23) have a 4. T_min is the beginning time in seconds of the identification. F_min is the minimum frequency in hertz of the identified call. T_max is the ending time in seconds of the identification. F_max is the maximum frequency in hertz of the identified call. The dataset is complete and contains no null values.

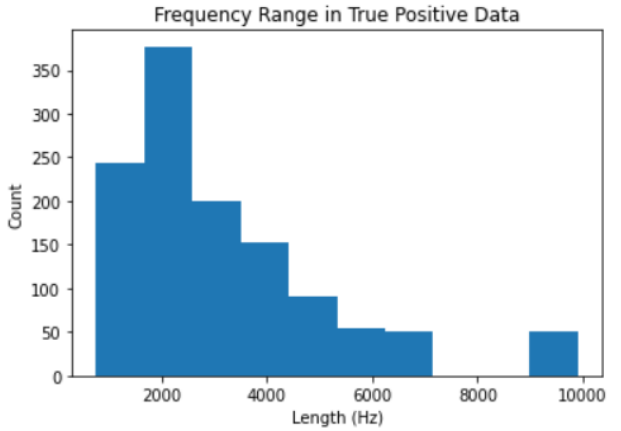
We investigated the distribution of each field and discovered some interesting results. The time lengths have a long tail distribution where most are under 3 seconds but they can be up to 8 seconds. This is useful because it means we are only using a few seconds of audio at a time so our model will run much faster than if the times were tens of seconds.

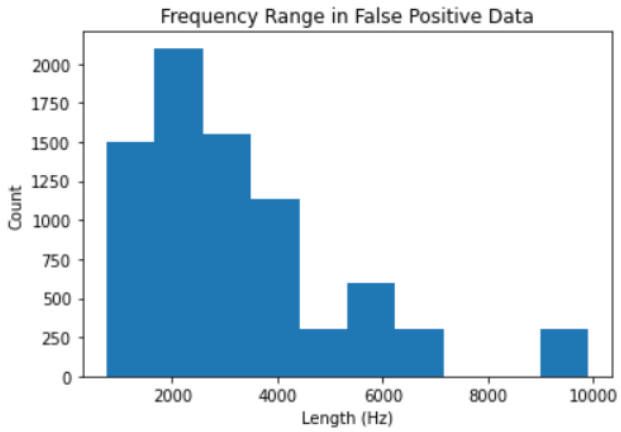




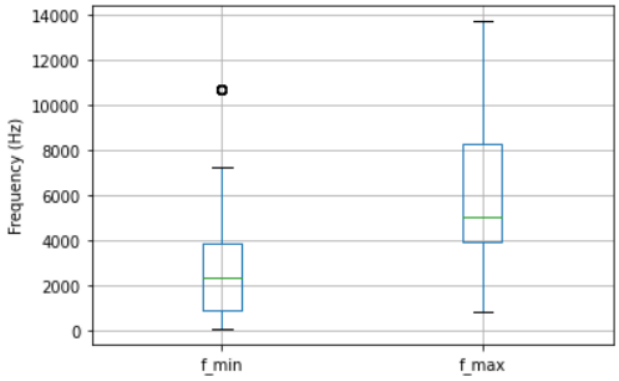
The frequency ranges which are the difference between the maximum and minimum frequency are pretty broad. The highest frequency is below 14,000 hertz and some samples range by over 9,000. Therefore, the frequency range is probably not too

useful since it is so wide.





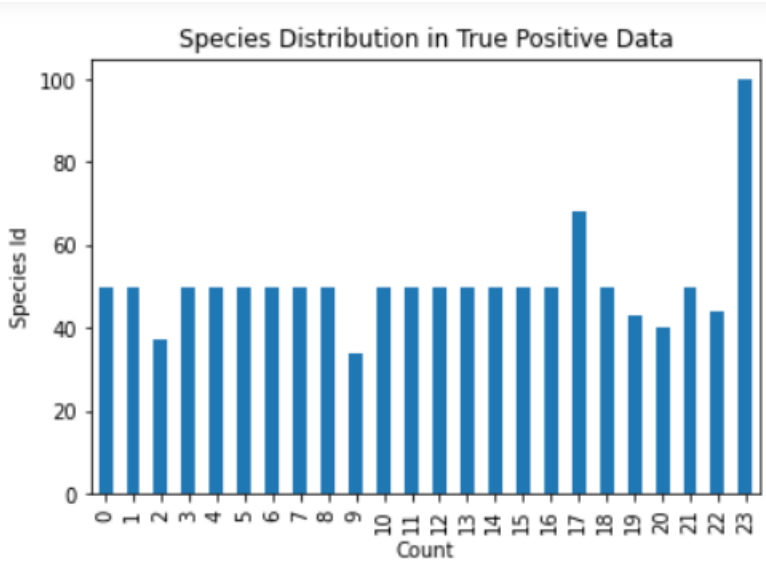
Most of this variation occurs based on the maximum frequency. The minimum frequency has a smaller distribution in comparison. We can use this knowledge to potentially focus on the higher frequencies as the distinguishing area for the species, since they will not overlap as much.

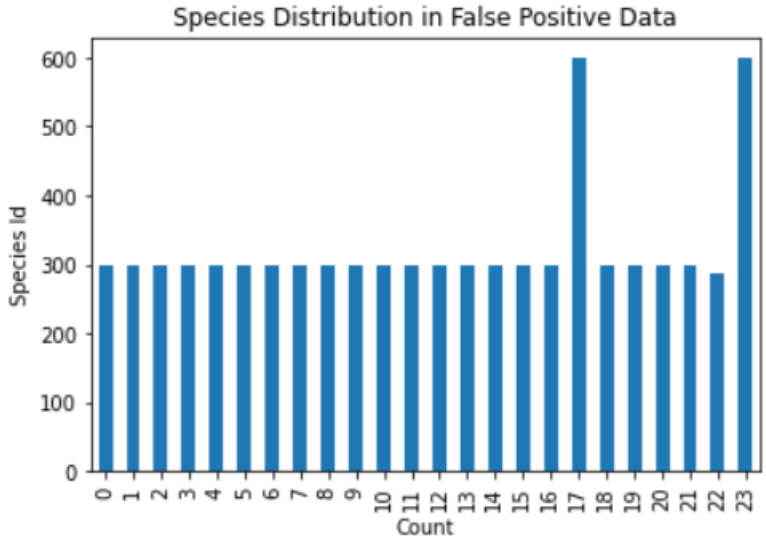


True Positive Frequency Box Plot

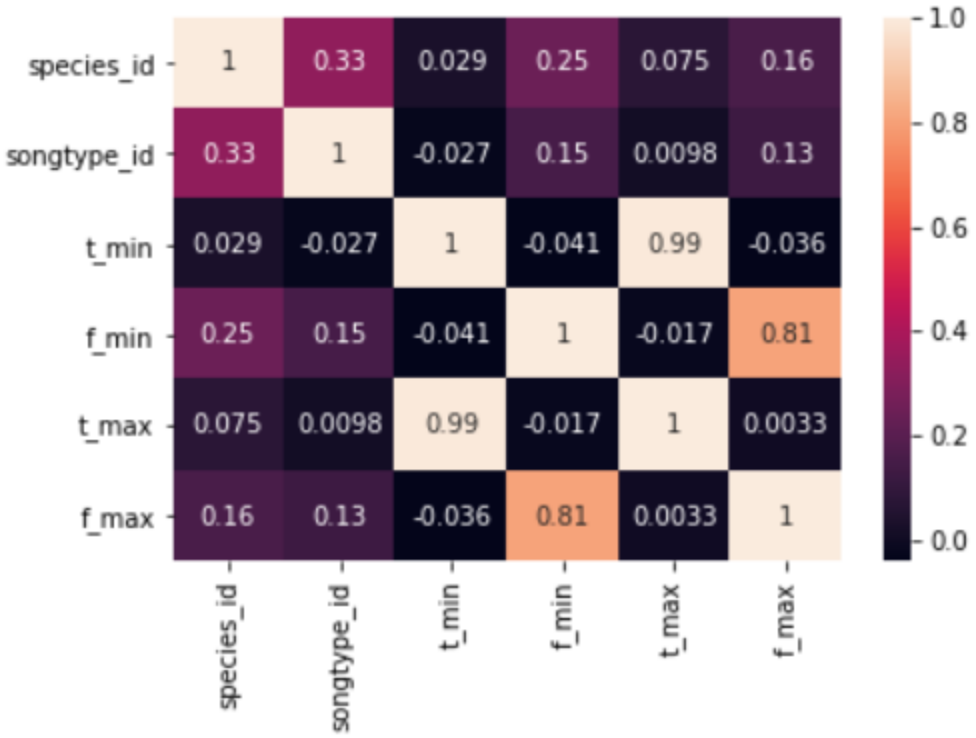
The data is generally balanced by species. The two upper outliers are the species with multiple song types. When you account for the two song types, each song type has

around the mean amount of data. The true positive also has some species that are slightly less present. The false positives are almost perfectly balanced.

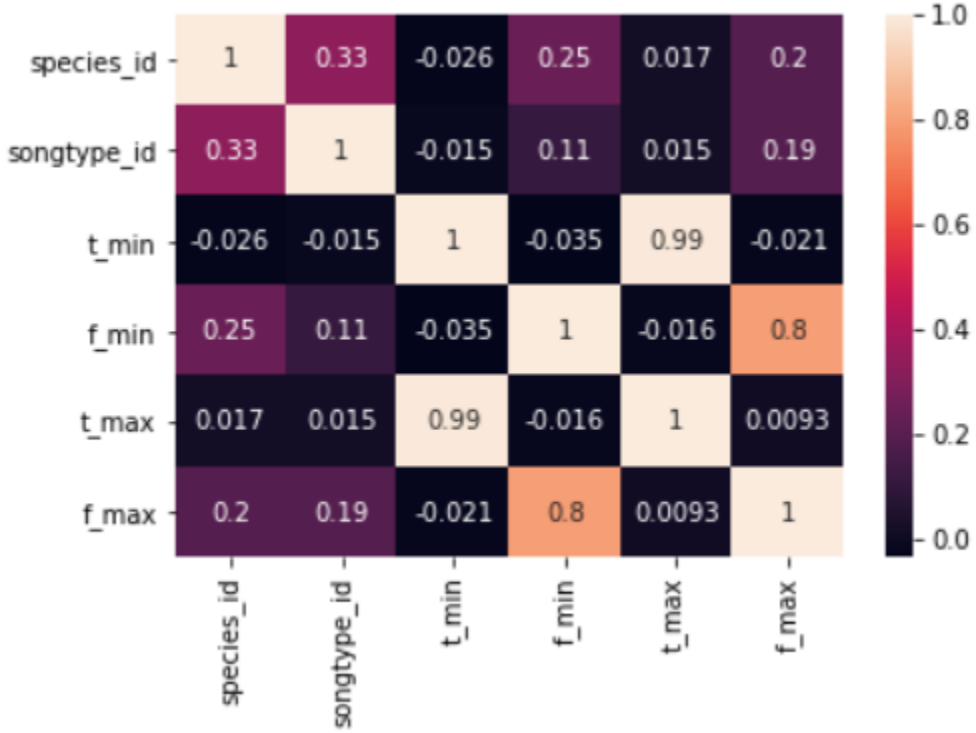




Running a correlation matrix on the columns reveals nothing unexpected. The only areas with a strong correlation are those that we assumed would be beforehand. Minimum frequency and maximum frequency are correlated, and minimum time and maximum time are correlated. Since so few species have multiple song types and there are only two possible values for it, song type and species are fairly strongly correlated. Species and frequency are also fairly correlated. This further supports our idea of using differing frequencies as a way to differentiate these species.



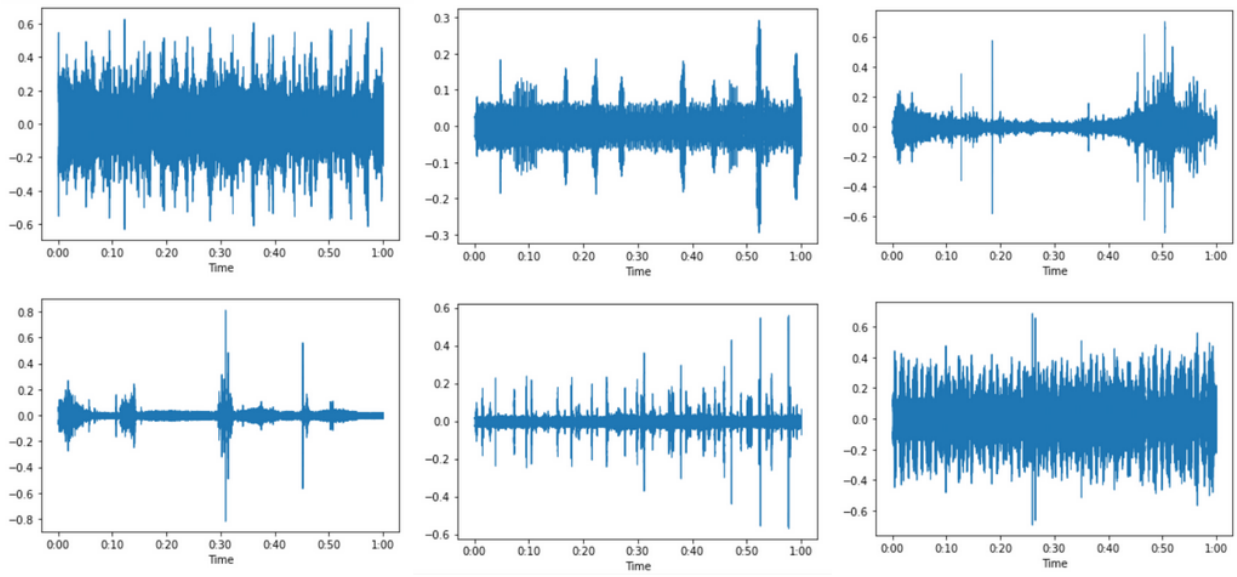
True Positive Correlation Matrix



False Positive Correlation Matrix

Audio Analysis

The audio recordings are 48000 Hz sampling rate and 60 seconds long. By looking at a few waveforms, the audio data contains huge variations in background noise, amplitude, and volatility. This variation is a huge issue because the model will have a hard time learning the defining characteristics.



Discussion

The most challenging part of working with this data is its incompleteness. An average audio file will have roughly one positive identification and two negative identifications, each only a few seconds long and for only one species. In comparison, the audio file is 60 seconds and there are 24 species. An innovative way of dealing with this issue is key.

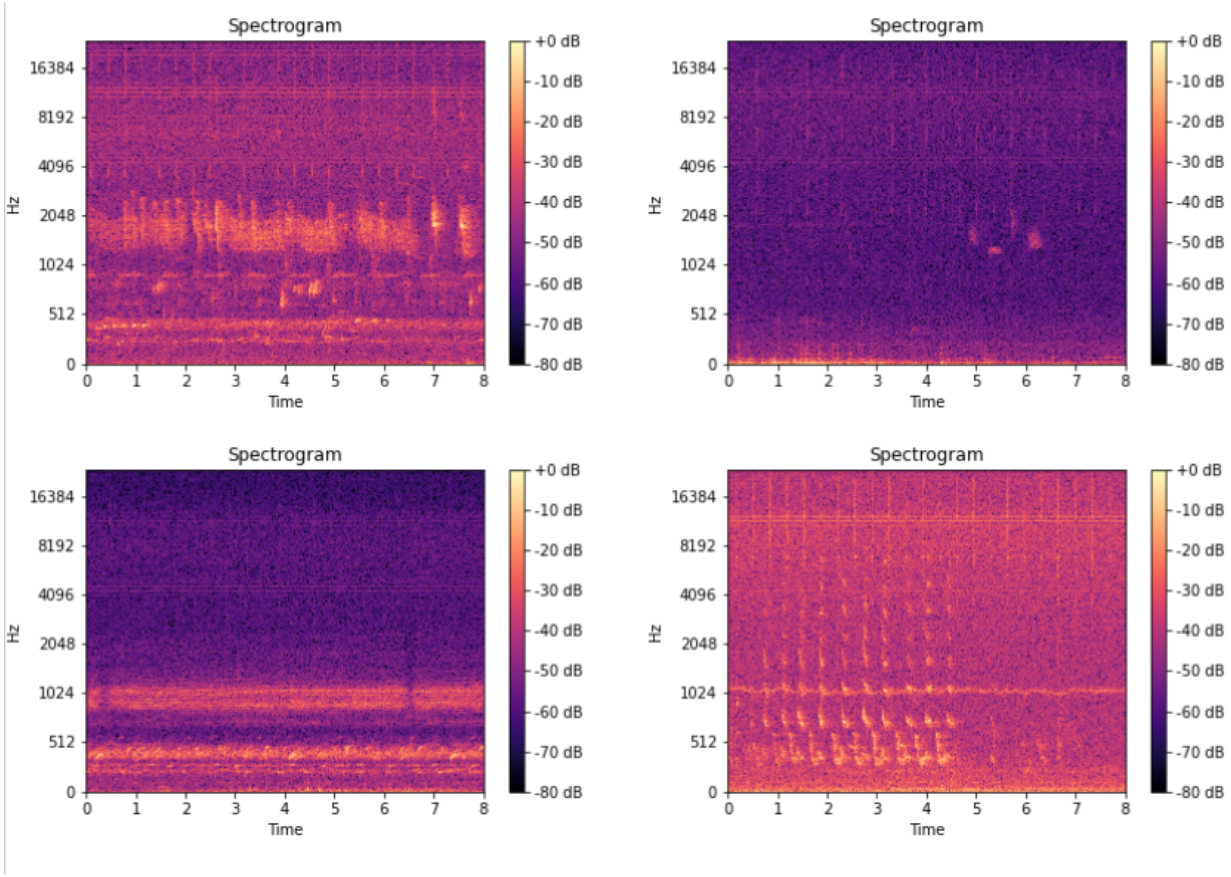
Some of the other challenges are the large amount of false positive data, the minimal non-audio information, and the unevenness in the audio recordings background noise and amplitude.

We are focusing on dealing with the audio issues through some innovative preprocessing. We will handle the data incompleteness in the development of our model.

One advantage of this data set is that the samples are very balanced. We will not have to worry about our model being good on some species and bad on others.

Preprocessing

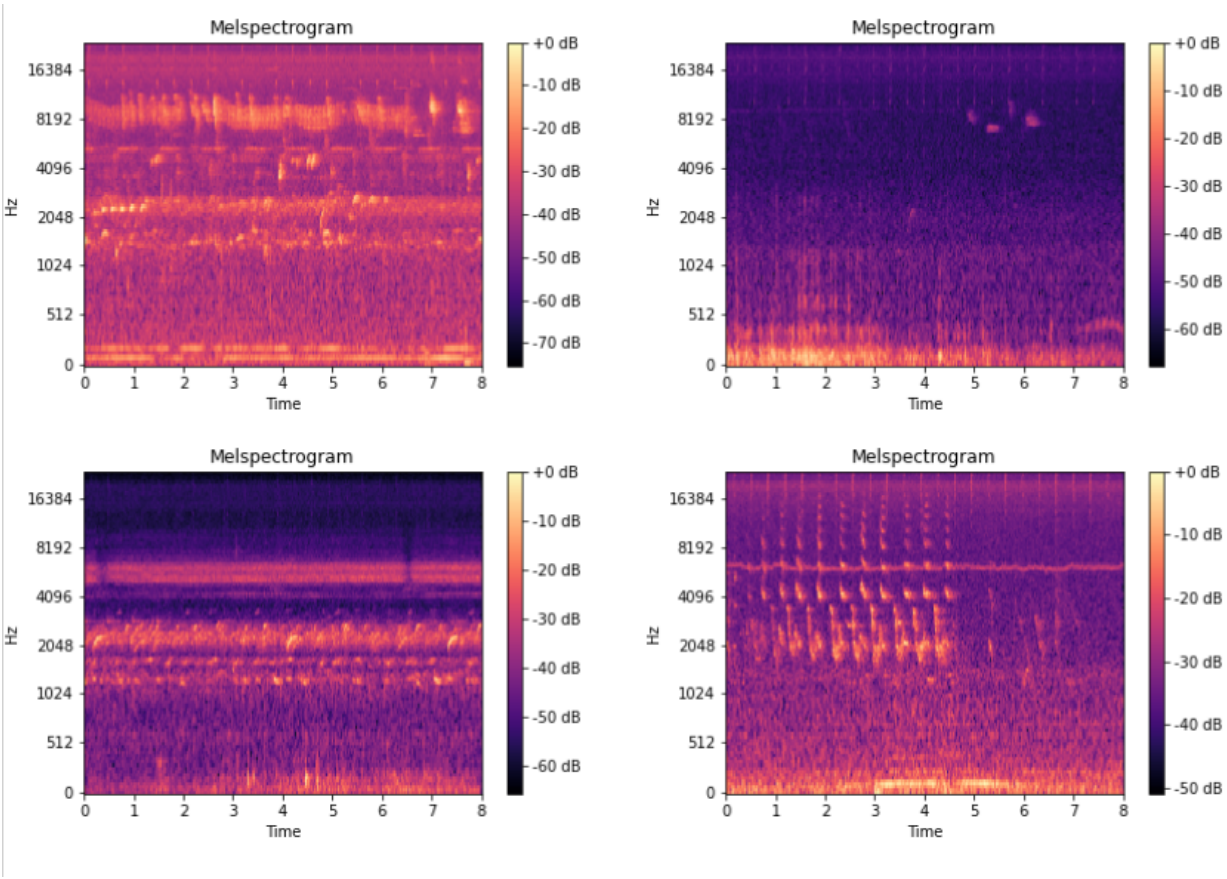
The metadata was very well maintained in its current state, but the audio data is difficult to work with for traditional machine learning methods, so we performed a lot of preprocessing on it. We began by isolating 8 second sound clips centered around the identification. 8 seconds was chosen because it is the maximum length of a sound clip identification. Then we created a spectrogram using a short time Fourier transform. This is a standard technique for creating a visual representation of audio. Using a fixed time means all the images are the same size, so image identification is more consistent on them.



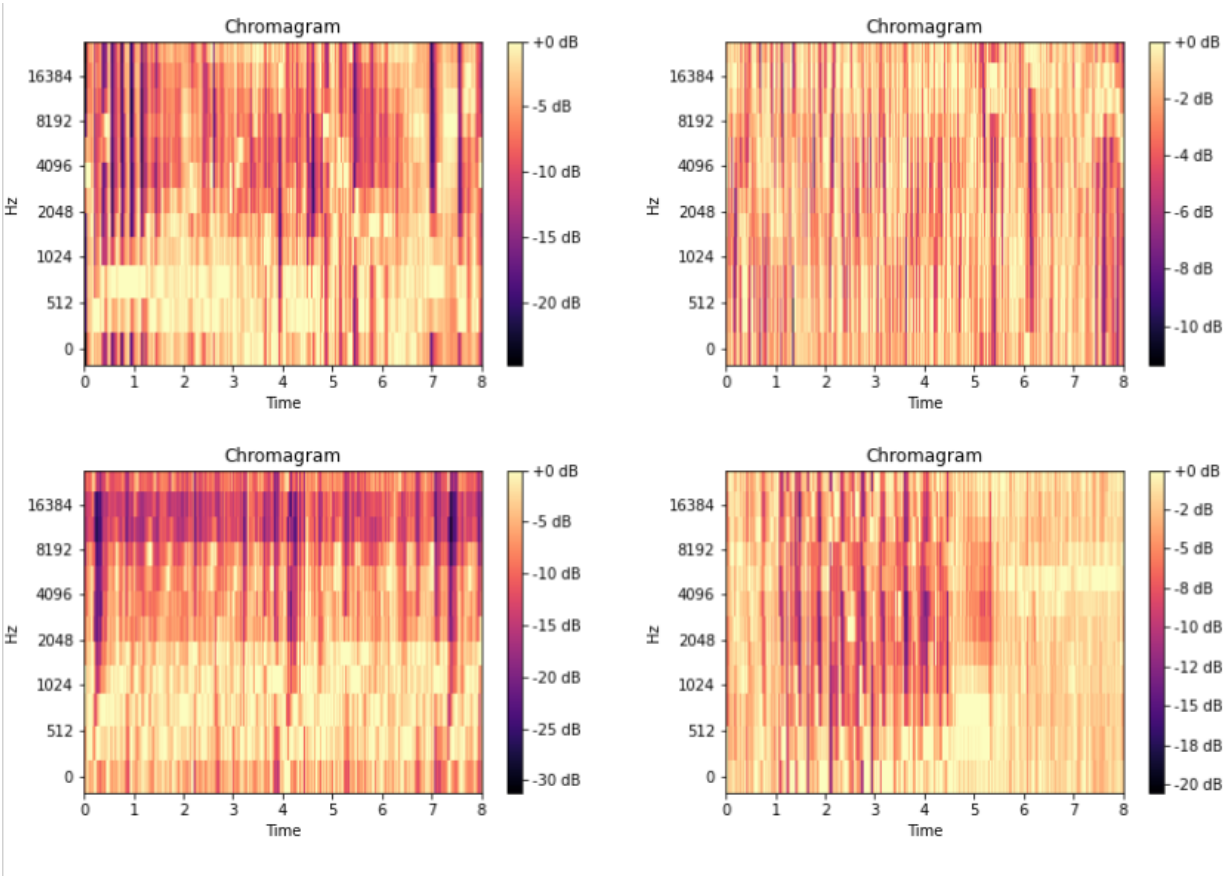
Sample Spectrograms

In addition to the spectrogram generation, we also created mel-spectrograms and chromagrams. Mel-spectrograms apply a mel filter to a spectrogram. A mel filter is a formula to match audio to human hearing capabilities. Higher frequencies are less distinguishable and quieter compared to low frequencies. A chromagram sums each of the 12 pitches at different octaves, to provide a composite image that displays the intensity of each pitch. We chose these methods because we thought they would each provide a different view of the data. A spectrogram will be better for looking at the higher frequencies like we discussed earlier. A mel-spectrogram would provide a visualization

closer to human hearing and by extension animal hearing. Since calls are made for other animals, this might be easier to distinguish calls through. The chromagram was chosen because bird species can often make a call at the same pitch but a different frequency, so a chromagram would better represent this difference.



Sample Melspectrograms



Sample Chromagrams

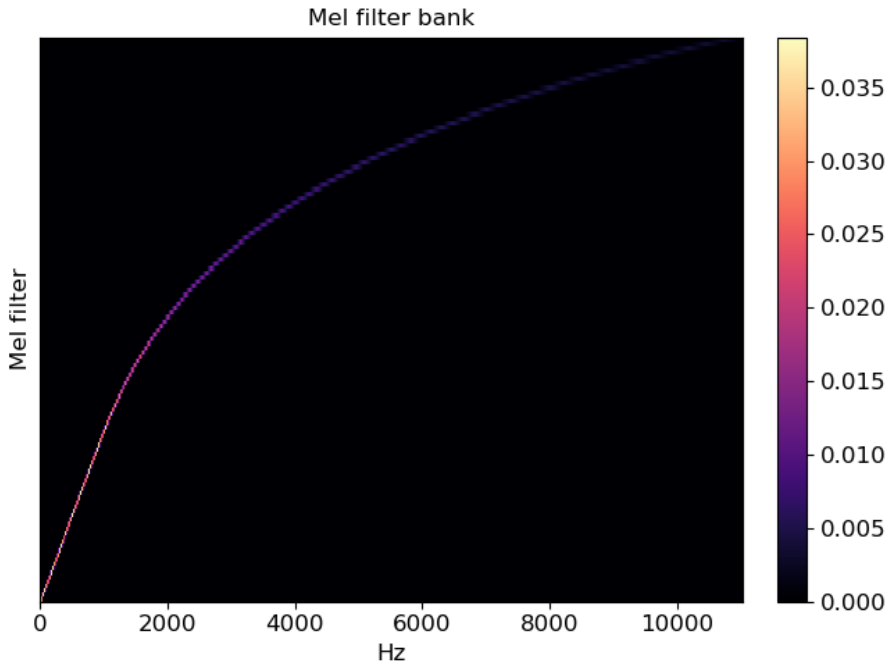


Image from [Librosa mel filters documentation]

After generating the graphs, we clipped the data to match the minimum and maximum frequencies found in the total dataset. This proved to be a minimal change because the minimum frequency was already 0 and the maximum frequency was cut from around 18000 Hz to around 14000 Hz.

We then decided we needed more data, so we came up with a method of increasing our data size. We added a normal distribution of noise to all our data and appended this data. We experimented with various amounts of noise, and decided to use both 0.2% and 0.5% of the maximum frequency. In each model, we trained using both, neither, and one or the other. Another idea we implemented to expand our data size, was to split the identifications into two clips. In each clip, we used 75% of the original identification size. This meant each clip had 50% overlap with the other. We found that this background noise addition and clipping gave us a huge jump in performance without too much of a concern of overfitting.

One final technique we tried was Discrete Wavelet Transform. Discrete Wavelet Transform allows the extraction of the key features of an image. We decided to perform DWT on both the spectrogram and melspectrogram. Because of the summation of different frequencies on chromagrams, chromagrams are already fairly pixelated and so the DWT components would have been less unique. While multiple levels of decomposition can be applied, we noticed a considerable enough difference after one level that we decided not to apply more. Some of the models we use can take the

decomposition as an input, so we will train it on both the original image and the decomposition and see which produces the better results.

Future Work

Given more time, we want to experiment with other methods of adding background noise and develop a more analytical way to see the tradeoff between overfitting and performance from our techniques to increase the dataset size. We are also interested in ways to remove background noise as an alternative. This would help standardize our dataset which suffers from huge variations in noise.