Signals and Systems Lab

BIRD SOUND DETECTOR

Syed Sameer Nadeem – sn02902 | Areej Al Medinah – aa02253

April 11, 2018

Contents

Introduction

There are millions of species of birds in the world, and each of them have a distinct sound. Taking advantage of the in-depth research of audio signals, it has been identified that the birds sounds have a number of distinct characteristics, such as pitch, amplitude and pattern quality which can be used to identify them.

The aim is to detect the species of birds in an environment using the audio acquired from the place, including environmental sounds as noise. Practical applications of such systems include identifying birds in their habitat, feeding birds according to their specie requirements and observing migration of birds from their original habitat. This can be used by environmental engineers, hobbyists as well as city development planners to see the presence of birds in a region.

Theory

The distinct MATLAB function used to identify unique bird sounds is the cross correlation function: **`xcorr()`**

Cross-correlation is a measure of similarity of two waveforms, also known as a sliding dot product or inner-product. For discrete signals, the cross-correlation is defined as:

$$r_{xy}[l] = \sum_{n=-\infty}^{\infty} x[n]y[n-l]$$

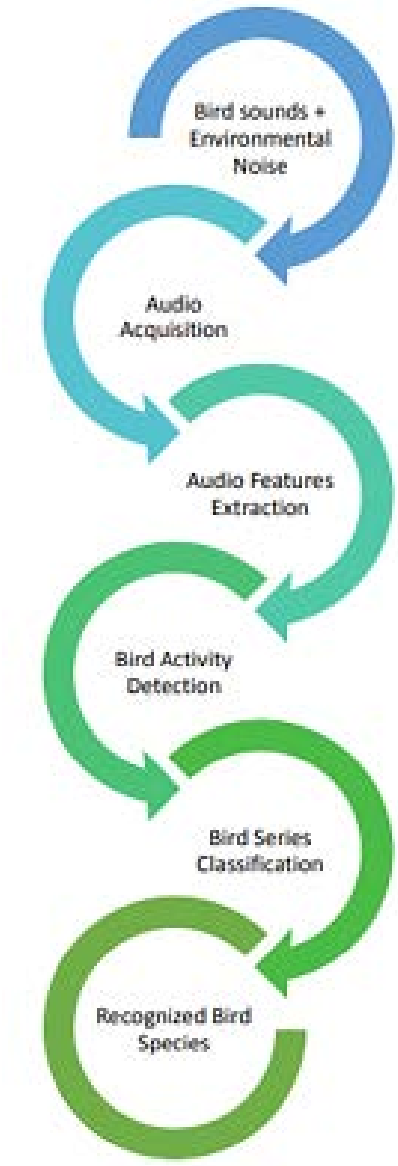
In this expression, the output of cross correlation is a function of l, which is the lag of the signal. The correlation sequence will be a function of lag which means at different lags we will have different degree (percentage) of similarity between the signals. In MATLAB function **`xcorr(signal_x,signal_y)`**, the lag by default is **`2N - 1`**, where **`N`** is the length of x or y, whichever is greater. If x and y have different lengths, the function appends zeros at the end of the shorter vector so it has the same length, **`N`**, as the other.

The function **`xcorr(signal_x,signal_y,maxlag)`** can process cross correlation with a specified lag of the range **`-maxlag to +maxlag`**.

Detection in signals uses correlation in significant applications. For example, if several communication signals are sent over a single channel simultaneously, they can be detected by cross correlation with the original functions. This enables the use of radar and sonar equipment in environments with abundant noise. Correlation is also used in image processing, such as to clear out noise from an image signal for clearer quality.

Methodology

The methodology adopted is demonstrated in the flowchart below:



.

First, the input audio signal is taken from the environment. This has sounds of birds in different habitats, along with environmental noise. Then, we correlate it with a set of pre-loaded signals of untampered bird sounds.

The cross correlated outputs are stored in an array, and the quantity of similarity is determined. This is determined by finding the number of signal-values above a certain

threshold magnitude (65% of the maximum). The output with the least number of these above threshold values has the highest similarity, as observed through the graphs as well.

After determining the best-fit from the correlated outputs, reverse engineering is applied to find out which original bird sound was correlated with the input, and that is the detected bird. Which is named and also the bird sound is played.

Implementation

Generating Input Signals:

Input signals were generated on Audacity by combining sounds of multiple birds, leaves, winds, etc. Some of them were also convoluted with tunnel and forest impulses to give it the effect of multiple habitats. Input signals were also acquired from open source files on the internet.

Loading Back End Data:

Six bird audios were downloaded from open source online sources. These were pre-loaded into the MATLAB software in an array structure. Each of them was also given the label of the bird's name to present as the final output as the detected bird.

Running the Software through Cross Correlation:

The software was initiated by calling the **`birdDetect(filename)`** function where the filename was of the input audio signal. The six correlated outputs were plotted on six graphs. Each of them had a unique pattern but one of them is expected to have a pattern very similar to the output of auto-correlated plots i.e. a spike of data at a particular point which is much larger compared to the of the data points. This is because of the similarity of signals of the original bird sound and the same bird's sound in the input signal.

Detecting the Bird Sounds through Graph Analysis:

The graphs were then analyzed for quantifying the similarity between the two waves. This would make the software produce a definitive output for the detected bird.

Playing the Bird Sounds:

For clearer design purposes, the input signal and the detected bird's sound is played, to verify the output through intuitive judgement of hearing and to let the user know of the bird detected.

Discussion

Over the course of this project, we came across many ways to go forward with it. Research had been done in the detection of birds via their calls by seeing patterns in the characteristics of their calls. Such as analysing their frequencies, the pattern of their call (ups and downs of amplitude) etc. The basic method used is the correlation of the input signal with the preloaded isolated calls of the birds. This we overtook. The rest can be implemented as further checks.

The main issue that we face using this approach is that many birds are closely related and have very similar calls. So, just a correlation check was proving to be futile here.

Once the basic correlation was done to detect the dominating bird sound in the signal, we tried to work towards detecting the habitat or background the bird was in. For this we tried to filter out the frequencies of just the background but in the process, data was being lost and effective correlation was not possible. We at the end tried to make a GUI for the software but were unable to complete it.

Our software, all in all is able to differentiate and detect many distinct birds via their calls.

Conclusion

The MATLAB software for detecting bird sounds was implemented with the ability to detect sounds of multiple birds present in its database. This was achieved through analyzing the outputs of cross correlation of the input signal with the database. The software, in its limitations, was working with accuracy.

MATLAB CODE

```
function [result,rcall, rcorr,inp] = birdDetect(filename)

[sound1,Fs1] = audioread('Bald Eagle.wav');

sound1 = sound1(:, 1);

[sound2,Fs2] = audioread('Great Tit.wav');

sound2 = sound2(:, 1);

[sound3,Fs3] = audioread('Pheasant Cuckoo.wav');

sound3 = sound3(:, 1);

[sound4,Fs4] = audioread('black capped chickadee.wav');

sound4 = sound4(:, 1);

[sound5,Fs5] = audioread('Hill Pigeon.wav');

sound5 = sound5(:, 1);

[sound6,Fs6] = audioread('Yellow Grosbeak.wav');

sound6 = sound6(:, 1);

Fss = [Fs1 Fs2 Fs3 Fs4 Fs5 Fs6];

birds = {'Bald Eagle' 'Great Tit' 'Pheasant Cuckoo' 'Black Capped
Chickadee' 'Hill Pigeon' 'Yellow Grosbeak'};

calls = {sound1,sound2,sound3,sound4,sound5,sound6};

comps = {[],[],[],[],[],[]};

least = [0 0 0 0 0 0];

[input_sound,fs] = audioread(filename);

disp('The input sound is')

sound(input_sound,fs)

pause(length(input_sound)*1/fs)

input_sound = input_sound(:,1);


for i = 1:6

    comps{i} = xcorr(calls{i},input_sound);
```

```
    subplot(2,3,i)
    plot(comps{i})
    title(['Input correlated with ' birds{1,i}])
    threshold = 0.65*max(comps{i});
    least(i) = length(find(comps{i}>threshold));
end
[~, index] = min(least);
result = ['This sound belongs to the bird named: ' birds{index}];
rcall = calls{index};
rcorr = comps{index};
inp = input_sound;
sound(calls{index},Fss(index))
figure()
subplot(2,1,1)
plot(comps{index})
title('The auto correlated')
subplot(2,1,2)
plot(input_sound)
end
```