

2011

# Nocturnal bird call recognition system for wind farm applications

Selin A. Bastas  
*The University of Toledo*

Follow this and additional works at: <http://utdr.utoledo.edu/theses-dissertations>

---

## Recommended Citation

Bastas, Selin A., "Nocturnal bird call recognition system for wind farm applications" (2011). *Theses and Dissertations*. Paper 521.

This Thesis is brought to you for free and open access by The University of Toledo Digital Repository. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of The University of Toledo Digital Repository. For more information, please see the repository's [About page](#).

A Thesis

entitled

Nocturnal Bird Call Recognition System for Wind Farm Applications

by

Selin A. Bastas

Submitted to the Graduate Faculty as partial fulfillment of the requirements  
for the Master of Science Degree in Electrical Engineering

---

Dr. Mohsin M. Jamali, Committee Chair

---

Dr. Junghwan Kim, Committee Member

---

Dr. Sonmez Sahutoglu, Committee Member

---

Dr. Patricia R. Komuniecki, Dean  
College of Graduate Studies

The University of Toledo

December 2011

Copyright 2011, Selin A. Bastas.

This document is copyrighted material. Under copyright law, no parts of this document may be reproduced without the expressed permission of the author.

An Abstract of  
Nocturnal Bird Call Recognition System for Wind Farm Applications

by

Selin A. Bastas

Submitted to the Graduate Faculty as partial fulfillment of the requirements  
for the Master of Science Degree in Electrical Engineering

The University of Toledo  
December 2011

Interaction of birds with wind turbines has become an important public policy issue. Acoustic monitoring of birds in the vicinity of wind turbines can address this important public policy issue. The identification of nocturnal bird flight calls is also important for various applications such as ornithological studies and acoustic monitoring to prevent the negative effects of wind farms, human made structures and devices on birds. Wind turbines may have negative impact on bird population. Therefore, the development of an acoustic monitoring system is critical for the study of bird behavior. This work can be employed by wildlife biologist for developing mitigation techniques for both on-shore/off-shore wind farm applications and to address bird strike issues at airports.

Acoustic monitoring involves preprocessing, feature extraction and classification. A novel Spectrogram-based Image Frequency Statistics (SIFS) feature extraction algorithm has been developed and was compared against traditional feature extraction techniques such as Discrete Wavelet Transform (DWT) and Mel Frequency Cepstral Coefficients (MFCC). Unlike traditional MFCC, signals were first cleaned with wavelet-denoising during the preprocessing stage. Additionally, a mixed MFCC-SIFS (MMS) technique was

also developed. Features extracted from proposed algorithms were then combined with various classification algorithms such as k-NN, Multilayer Perceptron (MLP) and Hidden Markov Models (HMM) and Evolutionary Neural Network (ENN). SIFS and MMS algorithms, combined with ENN and MLP, provided the most accurate results. Proposed algorithms were tested with real data collected during the spring migration, around Lake Erie in Ohio, of five nocturnally migrating bird species native to Northwest Ohio. Also, sparrows, warblers and thrushes passing over the University of Toledo, the Ottawa National Wildlife Refuge and Ohio State University's Stone Lab between April 20 – May 29 were calculated. Quantification of class level migratory bird has been presented. The developed bird flight call recognition system is suitable for deployment for both on-shore & off-shore wind turbine locations. This system would be suitable for 24/7 remote sensing.

# Acknowledgements

Firstly, I would like to thank my advisor, Mohsin Jamali, for giving me the opportunity to work in his research group. His help and guidance has helped me succeed during my graduate school career, and will continue to help me with my future endeavors. I would like to thank Dr. Junghwan Kim and Dr. Sonmez Sahutoglu for their time and serving as my committee members. I must also show my gratitude for Prof. Peter Gorsevski from the Department of Geospatial Sciences at BGSU, Prof. Joseph Frizado of the Department of Geology at BGSU and Prof. Verner Bingman of the Department of Psychology at BGSU for their guidance throughout this project, which was partially funded by the Department of Energy (Contract #DE-FG36-06G086096). I would also like to thank fellow students Mohammad Majid, Golrokh Mirzaei, Jeremy Ross for their important contributions to this work.

I am very grateful to have very special people in my life like Tugce Ayar, Ozlem Ozsoy and Richard Hausman who unremittingly support, help and be right next to me whenever I need.

I would like to dedicate this thesis to my family such that, without their support and endless love none of this would have been possible.

# Table of Contents

Acknowledgements.....	v
Table of Contents .....	vi
List of Tables .....	x
List of Figures .....	x
1 Introduction.....	1
1.1 Background and Previous Work .....	1
1.2 Contributions of the Research.....	5
2 Bioacoustics .....	7
2.1 Introduction to Bioacoustics Signals .....	7
2.2 Basics of Bioacoustics Digital Signal Processing.....	10
2.2.1 Concept of a Sound and Digitization .....	10
2.2.2 Representation of a Sound Signal .....	12
2.2.2 Discrete Fourier Transformation (DFT) and Spectrogram .....	13
2.2.4 Bioacoustics Signal Enhancement .....	15
2.3 Bird Sound Production and Vocalization.....	16

2.4 Analysis of Bird Sounds .....	20
2.4.1 Commercial Sound Analysis Systems .....	21
2.5 Bioacoustics Equipment and Software Used in This Work .....	23
2.5.1 Raven Pro 1.3 .....	23
2.5.2 Song Scope Software .....	25
2.6 Bird Species Used in This Study.....	27
3 Feature Extraction.....	32
3.1 Wavelet Coefficients.....	33
3.1.1 Continuous Wavelet Transform.....	33
3.1.2 Discrete Wavelet Transform and Multi-Resolution Analysis.....	35
3.1.3 Wavelet Families and Features .....	39
3.2 Wavelet-denoising Based Mel Frequency Cepstral Coefficients .....	42
3.2.1 Wavelet-denoising .....	42
3.2.2 Filtering and Normalization.....	48
3.2.3 Mel Frequency Cepstral Coefficients .....	49
3.3 Spectrogram-based Image Frequency Statistics (SIFS).....	54
3.4 Mixed MFCC and SIFS (MMS) .....	64
4 Classification.....	<b>Error! Bookmark not defined.</b>
4.1 K-nearest Neighbor Classifier (k-NN).....	66
4.2 Hidden Markov Model (HMM) Classifier.....	69



4.2.1 Discrete Markov Processes .....	70
4.2.2 Hidden Markov Models .....	70
4.2.3 K-Means Algorithm .....	76
4.3 Multilayer Perceptron (MLP) Classifier .....	78
4.3.1 Concept of Perceptron.....	78
4.3.2 Multi-Layer Perceptron.....	80
4.3.3 Backpropagation Algorithm.....	82
4.4 Evolutionary Neural Network (ENN) Classifier.....	85
4.4.1 Genetic Algorithm (GA) .....	85
4.4.2 ENN Algorithm.....	87
5 Data Collection and Simulation .....	93
5.1 Data Collection .....	93
5.2 Simulation .....	96
5.3 Experiment Setups and Results.....	98
5.4 Quantification of Class Level Migration In NW Ohio .....	107
6 Conclusions and Future Work .....	112
6.1 Conclusions.....	112
6.2 Future Work .....	113
References.....	115
Appendix A.....	125

Other Experiments Performance.....	125
------------------------------------	-----

# List of Tables

1.1 Summary of the recognition algorithms for the nocturnal flight call analysis .....	5
2.1 Frequency bands and range of the animal sound spectrum .....	8
2.2 Commercial systems and programs for bird sound analysis.....	22
2.3 Spring migration table of Northwest Ohio.....	28
2.4 Species, their classes and scientific names that are used in this work .....	29
3.1 Equations for CWT and DWT .....	35
4.1 Sigmoid function and its illustration.....	82
4.2 Basic steps of the Genetic Algorithm .....	87
5.2 Training and test calls that are used in this thesis.....	96
5.3 Comparison of the similarity measurements of k-NN algorithm.....	99
5.4 HMM configurations for flight call classification .....	101
5.5 ENN configurations for flight call classification .....	101
5.6 Percentages of classification accuracy for each feature extraction scheme.....	103
5.7 Results of Song Scope Implementation .....	106

# List of Figures

1-1 Basic bird flight call identification system .....	3
2-1 General block diagram of a bioacoustics signal detection system.....	7
2-2 Generalized block diagram of bird bioacoustics detection and recognition system.....	9
2-3 Pure sine wave before and after sampling.....	11
2-4 Time and frequency representations of a sine wave.....	13
2-5 System of a bird sound production .....	17
2-6 Spectrogram of Bewick's Wren song with basic hierarchical parts.....	18
2-7 Spectrogram representations of five different types of syllables .....	19
2-8 Block diagram of Song Scope Software.....	25
2-9 Batch processing snap-shot while common nighthawk.....	26
2-10 Spectrograms and photographs.....	30
3-1 Filtering process of discrete wavelet transform.....	388
3-2 Three level discrete wavelet decomposition.....	388
3-3 Morlet wavelet function.....	39
3-4 Scaling and wavelet functions of db 10 and sym 10 wavelet families .....	41
3-5 Wavelet denoising based MFCC feature extraction scheme .....	422
3-6 Original signal, Swainson's Thrush call, with noise .....	455
3-7 Spectral responses of the real signal.....	455

3-8 Detail coefficients of the signal .....	466
3-9 Approximation coefficients of the signal .....	466
3-10 Real signal, noise and denoised signal after wavelet denoising .....	47
3-11 Spectral responses after wavelet-based denoising.....	48
3-12 Block diagram for MFCC feature extraction method.....	5050
3-13 Mel spaced filterbank with 24 triangular bandpass filters.....	52
3-14 Mel scale vs. Hertz scale .....	533
3-15 Waveform and spectrogram of Swainson's Thrush call.....	555
3-16 3D spectrogram of Swainson's Thrush test call. ....	577
3-17 Spectrograms of Swainson's Thrush, Savannah Sparrow and Tennessee Warbler. .....	588
3-18 Features after cleaned spectrogram .....	61
3-19 Dimension reduction part of the SIFS feature extraction method .....	62
3-20 Summary of the SIFS Technique.....	633
3-21 Recognition block diagram with MMS feature extraction .....	655
4-1 k-NN classification for an unknown bird.....	68
4-2 Discrete-HMM model for bird call classification.....	788
4-3 Model of a simple perceptron .....	799
4-4 Linearly separable set .....	80
4-5 Linearly inseparable set .....	80
4-6 Three layer feedforward network .....	81
4-7 Block diagram of Evolutionary Neural Network .....	88
4-8 Single-point crossover operation .....	91

4-9 Mutation Operation.....	91
5-1 Wildlife Acoustics SM2 recorder .....	94
5-2 Google satellite view of the project area in Ohio, USA .....	95
5-3 Configuration of the SM2 recorder .....	95
5-4 Flow diagram of the overall system and MATLAB GUI.....	97
5-5 Snapshot of MATLAB GUI .....	97
5-6 Performance graph of MLP when MMS feature extraction scheme was used .....	102
5-8 Performance of Feature Extraction methods .....	104
5-9 Song Scope parameter and recognizer information .....	105
5-10 Performance comparison of the classifiers .....	106
5-11 Average number of total thrushes.....	108
5-13 Average number of total sparrows.....	108
5-16 Average number of total warblers .....	109
5-17 Average number of total thrushes.....	110
5-18 Average number of total sparrows.....	111
5-19 Average number of total warblers .....	111
A-1 Representation of hypothetical bird counting method.....	126
A-2 Amplitude vs. distance graph for hypothetical bird call counting.....	127
A-3 Simple demonstration of hypothetical bird counting method.....	128

# Chapter 1

## Introduction

### 1.1 Background and Previous Work

Identifying nocturnal bird flight calls is important for various applications such as ornithological studies and acoustic monitoring to prevent negative effects of human made structures and devices on birds [1] [2]. As stated by Scott Brandes, “birds are the most specious group of vertebrates on the planet [and] are important consumers at several tropic levels” [3]. The latter mentioned quality means that birds play an integral role in controlling the insect population [4], plant-seed dispersal [5], and even flower pollination [6]. Birds are not only vulnerable to climate change, but also to man-made habitat changes including interference by structures such as aircrafts [7], wind turbines [8] [9], electrical lines and towers [10]. Of specific interest to this study is the interaction of birds with wind turbines, which has become an important public policy issue [8]. Therefore, the development of an acoustic monitoring system is critical for bird preservation and the study of their behavior.

Birds primarily produce their sound through the use of a unique organ known as a

syrinx [11]. This organ is complex in structure and function and contributes towards broad spectrum of vocalizations that birds, as a whole, are able to produce [11]. These bird vocalizations can be divided into two categories: songs and calls [12]. However, not all birds are capable of producing songs [12]. Singing is limited to Passeriformes, or perching birds. This means that nearly half of the birds in the world do not produce songs [13]. This may initially seem problematic when considering an audio signal based recognition and classification method, however, most birds use vocalizations which are short and unmusical and can be termed as calls [12]. From this, it is obvious that species level bird identification should be based on calls rather than songs. Of particular interest to this study are species which are being affected by wind turbines. Due to the placement of these wind turbines, which is usually in wide, open fields, night migration birds are especially susceptible to interference by them. Conveniently, many bird species give flight calls during nocturnal migration [14] [15]. These calls provide an ideal medium for species level identification and quantification during hours of darkness.

Nocturnal flight call detection and recognition system is an application of bioacoustics signal detection and recognition systems. It consists of a flight call detection hardware system and a recognition software system [16] as shown in Figure 1.1.



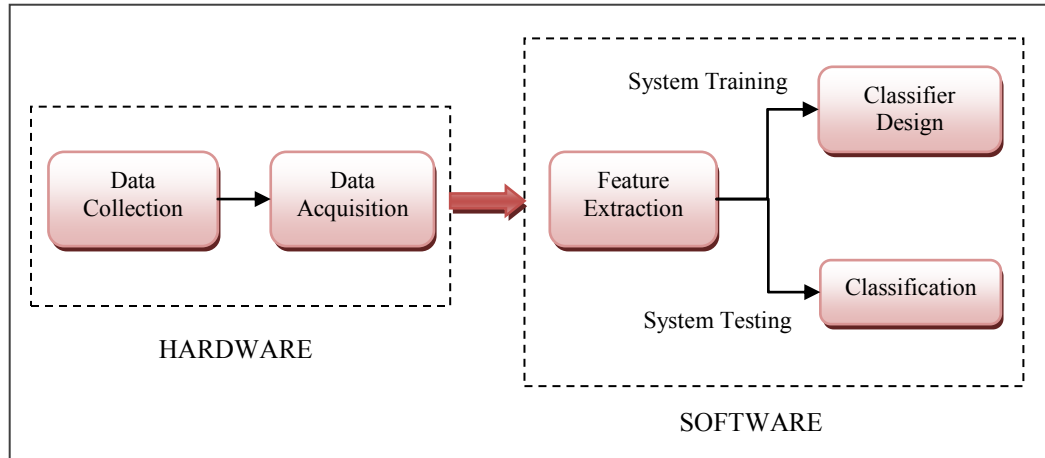


Figure 1-1: Basic bird flight call identification system

Basic components of the hardware system are microphones and recorders which collect and save bird calls. A flight call recognition system consists of feature extraction and classification. Recognition of bird species by their sounds is related to audio signal classification, speech recognition, music genre classification and pattern recognition, which have been widely studied [17] [18] [19]. Patterns of bird calls are parameters of sound events [20]. These parameters are known as acoustic features of sounds [20]. Success of recognition highly relies on how well sound events of calls are represented by features [21].

To date, many studies have been performed on bird species classification [22] [23] [24] [25]. Traditional feature extraction methods for acoustic features include frequency and time domain features [22]. Examples of the features used are spectrum, bandwidth, average energy, zero-crossing rate, Mel Frequency Cepstral Coefficients (MFCCs), wavelet coefficients and Linear Predictive Coding (LPC) coefficients [22] [24] [26]. Schrama et. al. extracted local seven acoustic parameters for the automatic detection of the European nocturnal flight calls which were: highest frequency, lowest frequency, call

duration, loudest frequency, average bandwidth, maximum bandwidth and average frequency slope [27]. Differing from MFCC, wavelet based features provide both frequency and temporal based information about the signal [26]. Selin et al used wavelet coefficients for bird species identification applications [25]. GK Verma used wavelet-based denoising at the preprocessing phase of MFCC for speaker recognition [26].

Previously, in the study of bird species classification, MFCC features were extracted and different classification algorithms, such as Dynamic Time Warping (DTW) [23], Hidden Markov Models (HMM) [23] Support Vector Machines (SVM) [22], k-Nearest Neighbor (k-NN) [24] were used. Also, Artificial Neural Networks (ANNs) have been widely used in the classification of birds, as well as other animal species [28] [29] [30]. Selin Arja et al. used ANNs such as unsupervised Self-organizing Map (SOM) and supervised Multilayer Perceptron (MLP) [25]. Results of these studies were encouraging and pointed toward the MLP being a better classifier than the SOM for recognition of bird sounds. Moreover, Evolutionary Neural Networks (ENN), which combine the feedforward neural network and Genetic Algorithm (GA), were used successfully for bat echolocation calls recognition [31].

Feature extraction and classification algorithms that are currently used for the nocturnal flight call recognition systems are given in Table 1.1:

Table 1.1 Summary of the recognition algorithms for the nocturnal flight call analysis

<b>Flight Call Recognition Techniques</b>	
<b>Feature Extraction</b>	<b>Classification</b>
Time domain Frequency domain Mel Frequency Cepstral Coefficients LPC Discrete Wavelet Transform Local Features	Clustering Methods Dynamic Time Warping Gaussian Mixture Models Hidden Markov Models Support Vector Machines Artificial Neural Networks

## 1.2 Contributions of the Research

This thesis proposes a nocturnal flight call recognition system for the migratory birds of Northwest America. Algorithms were developed on the MATLAB environment. For this, acoustic features are first extracted, via MFCC and DWT. In order to achieve more distinctive features for accurate classification, a novel Spectrogram-based Image Frequency Statistics (SIFS) is proposed and implemented. SIFS and MFCC features are then combined in a Mixed MFCC and SIFS (MMS) feature extraction scheme to obtain robust features for bird call identification.

Features with MFCC, DWT, SIFS and MMS schemes are used with a number of classifiers such as k-NN, Multilayer Perceptron (MLP), Hidden Markov Models (HMM) and Evolutionary Neural Network (ENN). A commercially available bioacoustics software, Song Scope, is also used for comparison purposes.

Above developed algorithm was used to quantify the number of thrushes, warblers and sparrows in different areas of northwest Ohio. For this, the total numbers of warblers, sparrows and thrushes passing over the University of Toledo, the Ottawa National

Wildlife Refuge and Ohio State University's Stone Lab between April 20 – May 29 were calculated. Results from these three locations were compared.

This thesis is organized as follows:

- Chapter 1: This chapter is the introduction to the thesis. It discusses the motivation behind the work. An overview of the thesis is provided to clarify the aim of this work.
- Chapter 2: Basics of bioacoustics and bioacoustics signals are explained. Aspects of flight call signal analysis system are introduced with a brief discussion on the digital signal processing and bird species.
- Chapter 3: Various feature extraction techniques that are used in this work are described. Explanations on the Mel Frequency Cepstral Coefficients (MFCC), Discrete Wavelet Transformation (DWT), Spectrogram-based Image Frequency Statistics (SIFS) and Mixed MFCC and SIFS (MMS) are presented.
- Chapter 4: The theory behind the classifiers, k-Nearest Neighbor (k-NN), Hidden Markov Models (HMM), Multilayer Perceptron Networks (MLP) and Evolutionary Neural Networks, are explained.
- Chapter 5: This chapter gives the details of data collection and simulation. The results of the performances of each recognition scheme are presented.
- Chapter 6: Conclusions and future work on this topic are given and suggested, respectively

# Chapter 2

## Bioacoustics

### 2.1 Introduction to Bioacoustics Signals

Bioacoustics is a multi-disciplinary science which studies animal sounds, including the production of the sound signal, as well as techniques used for its detection and recognition [32] [33] [34] .

A general block diagram of a common bioacoustics detection system is shown in Figure 2-1.

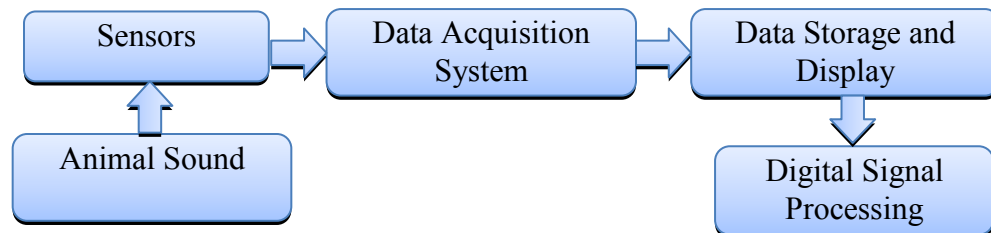


Figure 2-1: General block diagram of a bioacoustics signal detection system

A sound of interest is initially captured by a sensor system which converts the sound into an electrical analog signal. A Data Acquisition System (DAQ) digitizes the analog electrical signal which can then be stored in a Personal Computer (PC). Once the data is stored, Digital Signal Processing (DSP) techniques are applied to the signals for analysis, which includes noise reduction, feature extraction and classification [34]. This sound analysis helps in understanding behaviors and physiological meaning of the bioacoustics signal. The animal sound spectrum spans a broad frequency range, which can be seen in Table 2.1 [35].

Table 2.1: Frequency bands and range of the animal sound spectrum

<b>Type Band</b>	<b>Approximate Range</b>
Infrasonic	0 Hz - 16 Hz
Audio	20 Hz – 20 kHz
Ultrasonic	20 kHz – 160 kHz

For example, African elephants communicate with infrasound, bat echolocations are in the range of ultrasonic spectrum and birds produce audio sounds [35]. The bioacoustics signals used in this study were bird vocalizations. The frequencies of all bird vocalizations are between 200 Hz to 15 kHz, which, as previously mentioned, is in the audio frequency range. A block diagram of a typical bird sound recognition system is given in Figure 2-2.

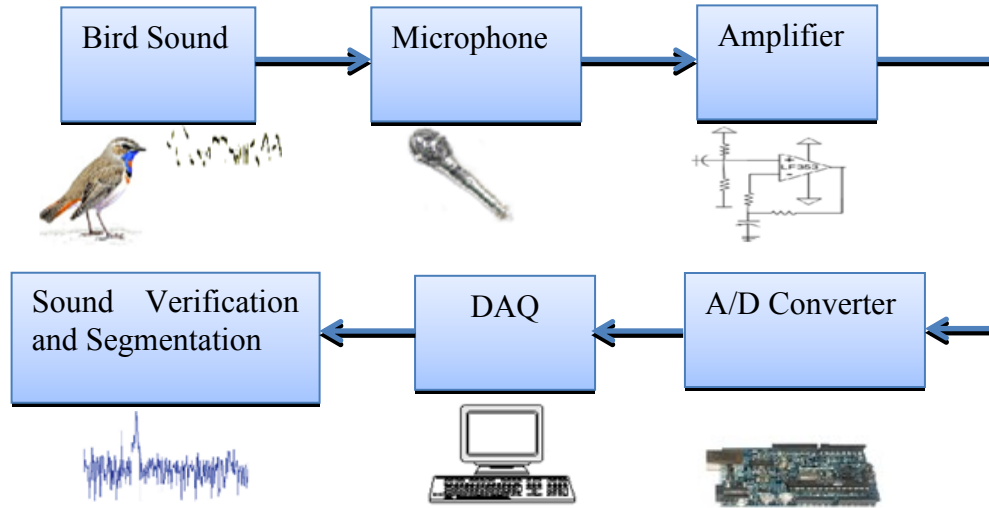


Figure 2-2: Generalized block diagram of a bird sound bioacoustics detection and recognition system

This work relates to nocturnal bird call recognition/quantification during migration. When identifying night flight bird calls, a suitable recording device must be chosen and placed in an appropriate field to capture the avian flight calls [36].

Recording devices include sensors to capture the sound of interest. The type of sensor, for the detection of bioacoustics signals, depends on the frequency range of the signal of interest. Since the frequency of the avian sound is in the range of audio signals, the most proper sensor is a microphone [36]. Microphones convert flight calls to electric signals [36]. This analog signal usually has a very low electric value and needs to be amplified [37]. To meet this need, some of the microphones have inbuilt amplifiers. An external amplifier must be used if the microphone does not have an amplifier [37]. Once the analog signal is amplified, it must be digitized using an Analog to Digital (A/D) converter [34]. The digital signal is then transferred to a PC with a Data Acquisition

(DAQ) system [34]. Finally, the digitized signal can be analyzed with sound analysis techniques [36]. Recognition systems, consisting of preprocessing, feature extraction and classification, can be utilized to recognize the incoming signal.

## **2.2 Basics of Bioacoustics Digital Signal Processing**

Signals represent a physical variable of interest as a function of time. The target bioacoustics signals of this work are bird vocalizations which are categorized as audio signals. Digital Audio Signal Processing is one of the digital signal processing applications [38]. A variety of PC hardware and software products are available to analyze an assortment of signals. Although these products change in capability, flexibility and complexity, some of the analytical techniques are similar in all systems. Commonly used operations in signal processing applications are convolution, filtering, and frequency-time domain conversions [39]. In this section, some of the basic concepts of audio signal digital processing techniques will be explained. This will provide a conceptual background on sound analysis and will help clarify feature extraction and classification techniques.

### **2.2.1 Concept of a Sound and Digitization**

Sound is created by sound sources which can be thought of as vibrating objects. Vibrating objects generate pressure waves which are formed by alternation of compression and rarefaction in an elastic medium such as air and water. The velocity of a sound wave through air is approximately 340 m/s at sea level [40]. Generated sound waves propagate from a source in all directions as air is pressurized. The amplitude of the pressure decreases with the square of the distance from the source [41]. Sound is recorded by a recording device, a microphone, which measures the changes in the air



pressure [36]. Microphones produce electrical signals, such as voltage and current, in proportion to the change in the air pressure. The rate of the air pressure vibrations is called the pitch of a sound and is often referred to as its frequency.

All bioacoustics signals coming from the recording device are analog in nature and vary continuously with time [40]. The varying electrical signals from the recording device are processed using an amplifier which is an analog signal processing unit [37]. However, before the sound signal can be analyzed by a PC, the signal must be digitized by an A/D converter, as PCs are based on a digital format [36]. A/D converters constantly sample the time-varying voltage of a continuous analog signal with a particular sampling rate [42]. Sampling rates are measured in Hertz (Hz) or samples per second. In another words, continuous information is converted into numerical values before processing it. To digitize a sound signal, the sound pressure, which was obtained from the output of a microphone, is sampled. Figure 2-3 shows an analog sine signal and its sampled version.

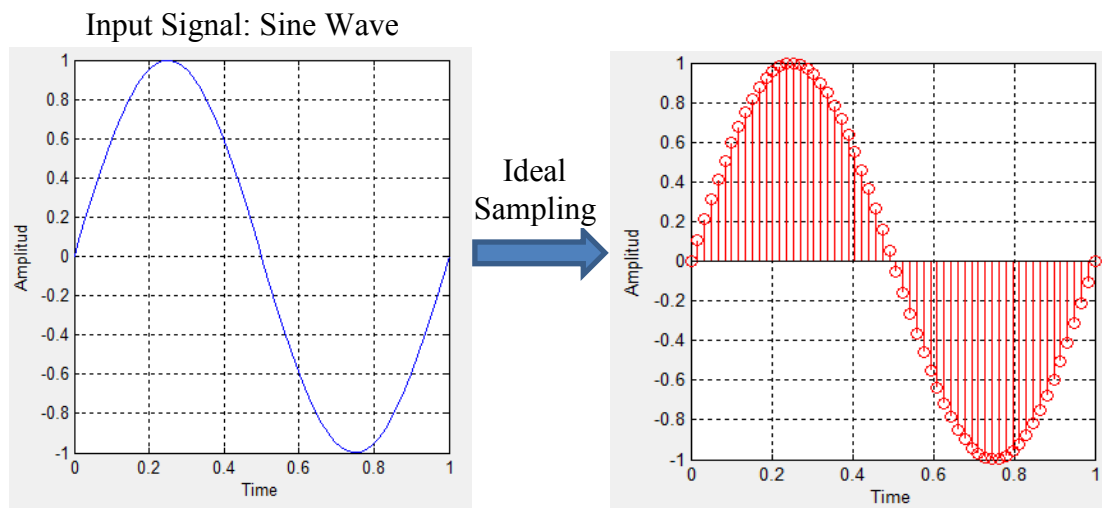


Figure 2-3: Pure sine wave before and after sampling

Analog signals are sampled according to the sampling theorem which states that the sampling frequency should be at least twice that of the highest frequency component in the signal [43]. If the sampling frequency is twice that of the highest frequency component of the signal, it is called the Nyquist Frequency [43] [42]. The Nyquist limit is mathematically expressed as:

$$F_s \geq 2F_c \quad (2.1)$$

where,  $F_s$  is the sampling frequency and  $F_c$  is the highest frequency component of the signal. For example, in order to detect high pitched sparrow calls (11 kHz), a sampling frequency of at least 22 kHz is needed.

### 2.2.2 Representation of a Sound Signal

A digitized sound signal can be represented in two different domains: the time domain and the frequency domain [44] [41]. Time domains represent the signal with a waveform plot in which the amplitude of the signal is a function of time. Amplitudes of the signal are the instantaneous air pressures of the bioacoustics signal. On the other hand, frequency domains represent the signal with a plot in which the amplitude is a function of the frequency. Frequency domains show how much of a signal's energy is present as a function of frequency and as the sound changes with time. Figure 2-4 shows the time and frequency domain representations of a simplest sound signal, pure sine, where the sampling frequency is 200Hz

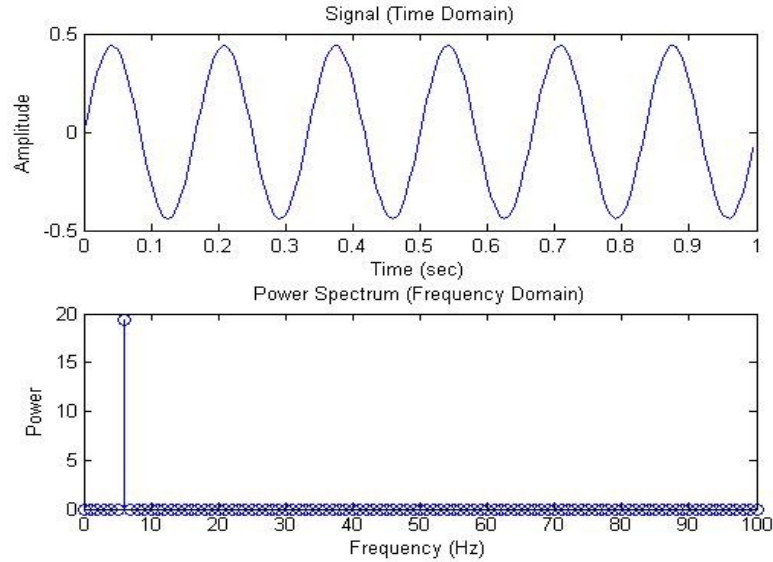


Figure 2-4: Time and frequency representations of a sine wave

It is seen from the frequency domain graph that the original frequency, which contains the information of the signal, is a vertical line. Frequency spectrums allow for the examination of energy in a frequency range of interest. This is also called spectral analysis [44]. Signals are transformed to the frequency domain from the time domain via Discrete Fourier Transformation (DFT) [39].

### 2.2.2 Discrete Fourier Transformation (DFT) and Spectrogram

Fourier transformation is a mathematical transformation operator that converts the time domain signal into the frequency domain [39] [43]. DFT is a type of Fourier transformation where the signal consists of discrete values. Since the signals are digitized by an A/D converter, DFT has more accurate transformation. Both inputs and outputs are discrete samples in DFTs. Assuming  $N$  is the length of a sequence,  $x(n)$

and  $0 \leq n \leq N - 1$ ; DFT is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad (2.2)$$

where  $k = 0, 1, 2, \dots, N - 1$ . DFT is a very important concept in digital signal processing.

Applications such as spectral analysis and frequency responses of filtering are based on DFT [39]. However, since there are totally  $N$  outputs, the evaluation of DFT needs  $O(N^2)$

operations [45]. Therefore, a computationally more efficient version of DFT is commonly used for such applications. This version is known as Fast Fourier Transformation (FFT). Results of FFT transform are exactly the same as those from DFT [39]. The only difference is the speed of algorithms, since FFTs need only  $O(N \log N)$  operations [45].

The most famous FFT algorithm was proposed by Cooley and Tukey in 1965 and is known as the Cooley-Tukey Fast Fourier Transform algorithm [46]. If DFT is assumed to have a composite size of  $N = (N1)(N2)$ , a Cooley-Tukey FFT algorithm recursively divides DFT into smaller sizes of  $N1$  and  $N2$  [45] [46]. In other words, the input of the

FFT is a sequence of complex discrete  $N$  amplitude values in time. The output of FFT transform is a sequence of  $N/2$  amplitude values which represents  $N/2$  discrete frequency components. In Fourier transforms, the frequency bins are all evenly spaced. Each frequency bin represents the frequencies at up to the half of the sampling rate ( $0 - Fs/2$ ).

In bioacoustics, the change in frequency information of the signal, with respect to time, is of interest. For instance, if a spectrum was created by applying FFT to an entire

bird sound, it would only show the representation of amplitudes of each frequency component. The variance of the amplitude values of different frequencies, rather than a general representation, is of interest. In order to obtain the FFT spectrum, the signal is divided into frames [43, 44]. Spectral leakage, which is also referred to as the sidelobe effect [44], is another issue with general FFT [47]. A time record is continuously repeated, and signals contained in this time record thus appear at periodic intervals that correspond to the length of the time record. If the time record has a non-integral number of cycles (non-periodic), spectral leakage occurs [47]. In other words, the non-integral cycle frequency component of the signal does not correspond exactly to that of the spectrum frequency lines. While this may be problematic, windowing the signal with window functions can eliminate spectral leakage [47] [48]. There are different types of window functions such as: Blackman, Hamming, Hanning, Rectangular, and Triangular [44]. Each of the window functions has different characteristics with different shapes. For instance a Blackman window has a -57 dB sidelobe rejection while a triangular window has -25 dB [44]. Therefore, each of the window functions reduces the sidelobe effect at different level.

#### **2.2.4 Bioacoustics Signal Enhancement**

One of the prominent problems in the application of bioacoustics signal identification is the noise. Noise can highly affect the classification results and cannot be totally removed [49].

The most common method to reduce noise artifacts in bioacoustics signals is the use of bandpass filters in which the frequency bands are limited to where the target signals

are seen [50]. For instance, flight calls of thrushes are in the range of 2 kHz to 4 kHz. Instead of analyzing the whole spectrum, with a bandpass filter, frequencies higher than 4 kHz and lower than 2 kHz can be filtered. Bandpass filters consist of low-pass and high-pass filters. The bandwidth of a bandpass filter is the difference between the upper and lower frequencies.

Another method for bioacoustics signal enhancement is spectral subtraction [51]. The additive noise model of a signal  $x(n)$  is expressed as:

$$y(n) = x(n) + d(n) \quad (2.3)$$

where,  $x(n)$  is the clean signal,  $y(n)$  is the noisy signal and  $d(n)$  is the additive noise. Spectral subtraction is based on this model. First, the noise spectrum of the signal is estimated by taking the Fourier transform of the clean regions in the waveform and calculating their magnitudes [52]. The frequency domain of each frame in the signal is estimated as:

$$\hat{X}(w) = [|Y(w)| - |\hat{D}(w)|]e^{j\phi_y(w)} \quad (2.4)$$

where,  $\hat{X}(w)$  is the estimated clean signal and  $\phi_y(w)$  is the phase component of the noisy signal.

## 2.3 Bird Sound Production and Vocalization

Humans produce sound through the use of the larynx, which is close to mouth. Unlike humans, birds have a different sound production organ: a *syrinx* [40]. The syrinx is deep

inside the chest of a bird and allows air, which is leaving the lungs, to pass through it. It is a bipartite vocal organ, indicating that it is capable of producing two tones simultaneously. Bird vocalizations can take the form of songs or calls, where songs are complex sounds which can be tonal or inharmonic and calls are shorter and less complex [12]. A basic schematic of the sound production system in a bird is shown in Figure 2-5.

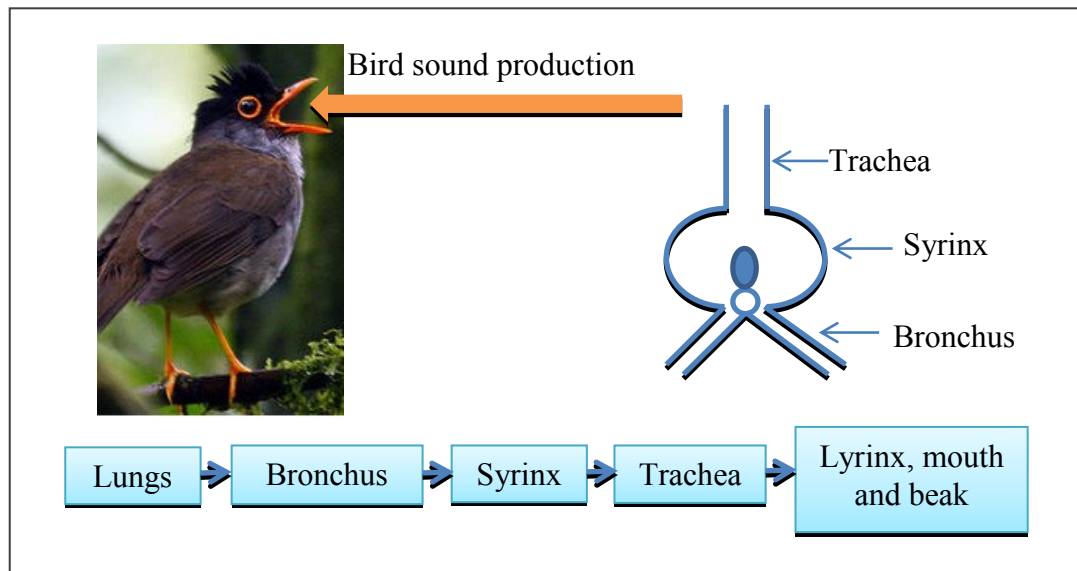


Figure 2-5: System of a bird sound production

Bird sounds are often represented by spectrograms. A bird sound consists of four parts: notes, syllables, phrases and songs [12]. Syllables can be composed of multiple notes, and the more notes a syllable has, the more complex its structure. Phrases occur as a series of syllables, occurring one after another, and songs are formed by sequence of phrases. Figure 2-6 shows a spectrogram of a Bewick's Wren's song with hierarchical parts.

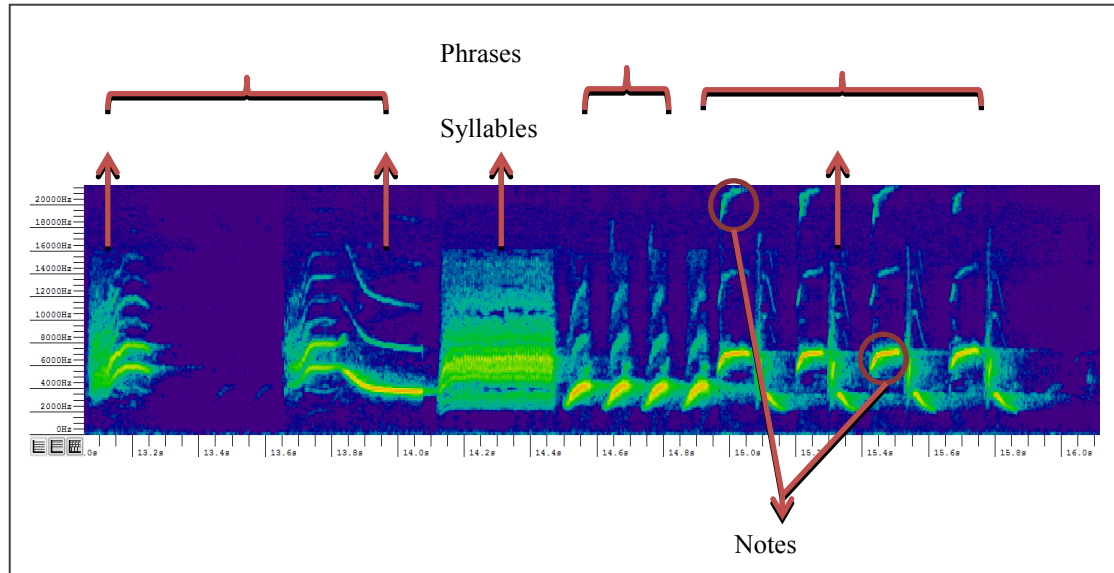


Figure 2-6: Spectrogram of Bewick's Wren song with basic hierarchical parts

The diversity of the sound units makes the implementation of bird sound identification systems challenging, especially at the species-level. This becomes more complex when the birds' sounds are combined with environmental noises. Furthermore, most of the individual species have more than one song and call. To make the species-level identification process more systematic, syllables are considered as they are the base of the bird vocalization [3]. Although there exists a wide range of different syllables, Figure 2-7 shows the basic types of syllables that are produced by five different species. From the first spectrogram to the fifth, the structures of the syllables are in the type of: constant frequency, frequency modulated whistle, broadband pulse, broadband with varying frequency components, and segments with strong harmonics [3] [53]. Harmonics are components of the waveform which have frequencies that are multiples of the lowest frequency components, or fundamental frequencies.



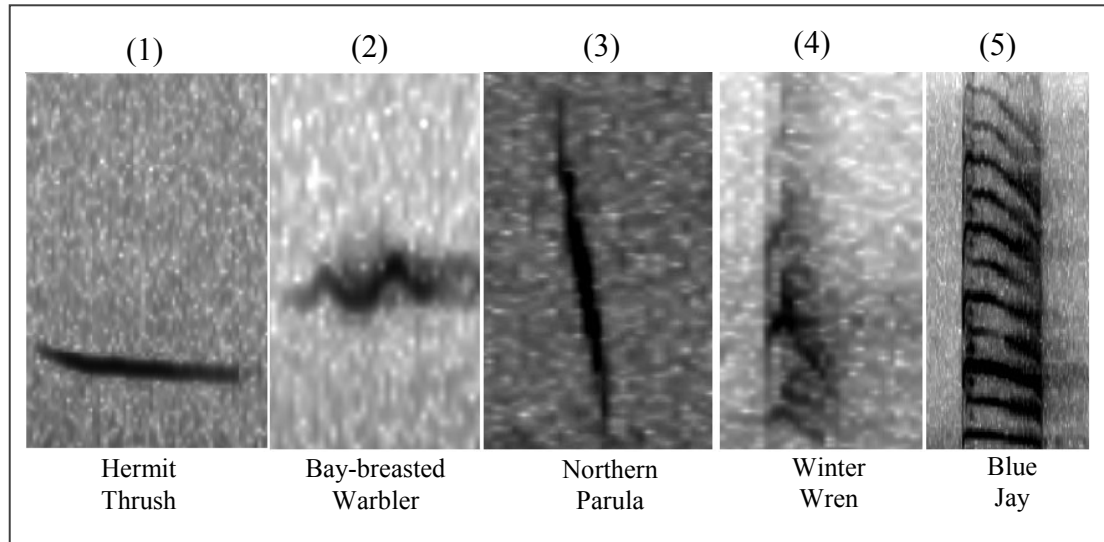


Figure 2-7: Spectrogram representations of five different types of syllables

Singing birds form only the half of bird population and are called *Passeriformes* [54]. While many birds do not sing, almost all of them use calls to communicate [54]. Generally, a call of a bird consists of one syllable or sequences of the same syllable. Birds usually have more than one call, such as alarm, territorial, begging, pleasure, flock, distress, nest and flight [54]. Different species have a different numbers of calls which usually varies between five to ten call types [54] [12]. Species can have one type of calls which may be similar to different type of calls in other species' calls [11] [12].

In this work, nocturnal flight calls were used for the identification of species. Flight calls are the main vocalization of many species and are used during long flights. However, not all species give flight calls [55]. Flight calls are used for flock coordination, especially during diurnal and nocturnal migration times [56] [54]. Flight calls are also used for foraging and interacting during breeding [57]. Also, in some species, such as the Swainson's thrush, flight calls are given during the day by perched

birds and at nights by birds in nocturnal migration flight. It should be stated that flight calls are not the only vocalization during flight. Some species sing and give chirp notes while in flight and some have no flight calls at all, not even during migration time [57]. Furthermore, some species, such as the yellow-rumped warbler, have more than one flight call.

Characteristics of flight calls change drastically from one species to another. Syllables that are shown in Figure 2-7 are all examples of flight calls. Ornithologists and bird watchers are mostly hear “zeep” and “seep” flight call sounds. Generally, flight call repertoires are generated during diurnal migration times since the birds must actually be seen [55]. Still, some of the nocturnal flight calls cannot definitively be said to come from a particular species and sometimes will be classified as “*complex*”[55]

## 2.4 Analysis of Bird Sounds

Sound analysis allows for the understanding of the characteristics of acoustic signals [34]. Although the identification of bird species, which is based on their sounds, is complex, it is more practical when only a limited number of species are being considered. Therefore, the first step is to prepare a list of species that are going to be identified. In this thesis, identification will be performed based on the flight calls. Therefore, different samples of flight calls for each species should be collected or found from reference guides. These flight calls are called training calls.

The analysis is composed of two main parts: flight call feature extraction and classification [3]. The types of features that will be extracted depend on the characteristics of the flight calls in the training calls database. Types of features and

classifiers commonly used for bird sound identification were given in Table 1.1.

A variety of analysis techniques and equipment are currently commercially available. Sound analysis algorithms are generally based on Digital Signal Processing (DSP). Also, aside from commercial products, new algorithms and systems can be developed through the use of software such as MATLAB. As it was stated before, spectrographic analysis represents the signal graphically and allows frequency components of the signal of interest to be examined or compared with other signals in order to find the similarities and differences.

#### **2.4.1 Commercial Sound Analysis Systems**

In commercially available packages, algorithms usually include noise reduction, feature extraction and classification techniques. All analyses are based on spectrographic techniques. Some commercial software, which can be used for bird sound analysis, are summarized in Table 2.2.

Table 2.2: Commercial systems and programs for bird sound analysis

Sound Editing Software	Analysis Software	Package System (Automatic Monitoring and Detection)
<ul style="list-style-type: none"> <li>-Adobe Audition</li> <li>-Audacity</li> <li>-Goldwave</li> <li>-Wave Flow</li> </ul>	<ul style="list-style-type: none"> <li>-Raven</li> <li>-Sound Analysis Pro</li> <li>-Sigview</li> <li>-Avisoft Bioacoustics</li> </ul>	<ul style="list-style-type: none"> <li>-Wildlife Acoustics: <ul style="list-style-type: none"> <li>• Song Scope Software</li> <li>• SM2 ARU</li> </ul> </li> <li>-SoundID: <ul style="list-style-type: none"> <li>• SoundID Software</li> <li>• SoundID ARU</li> </ul> </li> </ul>

Sound editing programs have tools to edit audio files visually. Tools, such as delete, copy, paste, cut, amplify, fade, normalize, reverse, echo, sound file conversion, export, and import are included in these programs. Analysis software provides visualization, measurement and analysis of sounds through the use of different algorithms. For instance, Raven [58] has various spectrogram parameters which can obtain high quality spectrograms. Sound Analysis Pro [59] has various feature values that can be viewed from the display.

Some of the analysis systems are package systems. A package analysis system consists of an Autonomous Recording Unit (ARU) and analysis software which are compatible to each other. The advantage of the package system is that sound can be automatically monitored and detected.

## **2.5 Bioacoustics Equipment and Software Used in This Work**

For this study, Cornell's Raven Pro 1.3 [58] and Wildlife Acoustic's Night Flight Call Package [60] were purchased. Raven Pro was used during the beginning phase of the study to visually examine the signals. Then, Song Scope Software and SM2 ARU were used for automatic monitoring and detection of night flight calls.

### **2.5.1 Raven Pro 1.3**

Raven Pro 1.3 [58] software is a product of the Cornell University Bioacoustics Research Program. Raven Pro acquires, measures and visualizes the bird sounds in waveform and spectrogram views.

The Raven recording software is capable of sending the audio input to a file while sequencing is specified by the user. It can record the real time data in Audio Interface File Format (AIFF) or Waveform Audio File Format (WAV) format. While recording, Raven allows the user to visualize the real-time signal with multiple views, such as spectrograms, waveforms, spectrogram slice views or selection spectrum views. Some important properties of Raven Pro 1.3 are:

- Raven has six types of window functions which are Blackman, Hamming, Hann, Kaiser, rectangular and triangular windows.
- Raven provides various measurements. Begin time, end time, low frequency and high frequency measurements are default measurements in Raven. In addition to the default measurements, based on the spectrogram values, the following are other available measurements that can be selected in Raven Pro: Average power, delta

power, energy, maximum frequency, peak frequency, maximum power, peak power, center frequency, 1<sup>st</sup> quartile frequency, 3<sup>rd</sup> quartile frequency, Inter Quartile Range (IQR) bandwidth, center time, 1<sup>st</sup> quartile time, 3<sup>rd</sup> quartile time, max amplitude, min amplitude, peak amplitude, peak amplitude, RMS amplitude, max time, min time, peak time, begin time, delta frequency, delta time, end time, high frequency, low frequency, length, max bearing, peak correlation and peak lag.

- Raven provides a paging feature when large data sets are involved. Also, Raven provides batch operations with its batch channel exporter.
- Raven Pro 1.3 is able to work with more than two channels and supports both NI-DAQ and Audio Stream Input Output (ASIO).
- Spectrogram correlation functions provide the peak correlation values which give the information regarding the similarity between spectrograms. Waveform correlation functions provide the time offset when two signals are the most similar to each other.
- There are three modes of default detection in Raven.

*Interactive detection:* Interactive detection is useful to visualize the performance of various detectors on a short selection of the recordings. Also, multiple detectors can be tested on a set of pages with color-coded results.

*Full detection:* With large data sets, the full detection mode is better than the other modes of detection. After the basic parameters are selected for short selections of a recording, the full detection can be performed over the entire recording.

*Real-time detection and the recorder:* This detection mode provides real-time detection of the events during the recording process. The detected events can be saved as individual sound clip files as well as in a recorded file sequence.

While Raven Pro 1.3 is superior to Song Scope in some aspects, the most significant disadvantage of it is the fact that it is not capable of classifying the bird calls efficiently. Also, its software recording feature is inefficient and inadequate for long periods of time.

### 2.5.2 Song Scope Software

Wildlife Acoustics has developed a commercial software program, called Song Scope [61], for the automatic detection of birds, bats and frogs. Song Scope is one of the few commercial software on the market for bird call recognition applications. The Song Scope software was used to classify both narrowband and wideband vocalizations with limited training data. The classification algorithm of Song Scope is based on Hidden Markov Models (HMM) and spectral feature vectors. The extracted features are similar to Mel Frequency Cepstral Coefficients (MFCC's).

Song Scope bioacoustics software can be used to scan large wav files. Song Scope provides viewable spectrograms and, more importantly, scans the long field recordings which can automatically classify species of interest. The basic steps of Song Scope are shown in Figure 2-8.

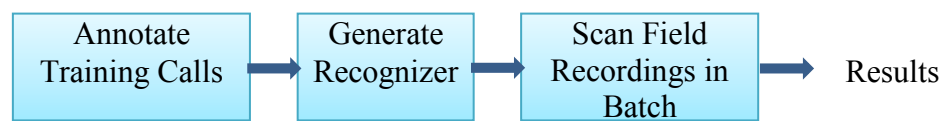


Figure 2-8: Block diagram of Song Scope Software

Implementation starts by determining the list of species that are of interest. For each species, a number of distinct training calls must be collected. Then, the recognizer of each species is built separately. In this step, all parameters are set manually depending on

the characteristics of the corresponding flight call. The constructed recognizers are then used to scan wav files. As a result, possible vocalizations are listed on a spreadsheet. Candidate vocalizations are tested manually and visually to see if they have been detected correctly. An example of a snapshot of the batch processing step is given in Figure 2-9.

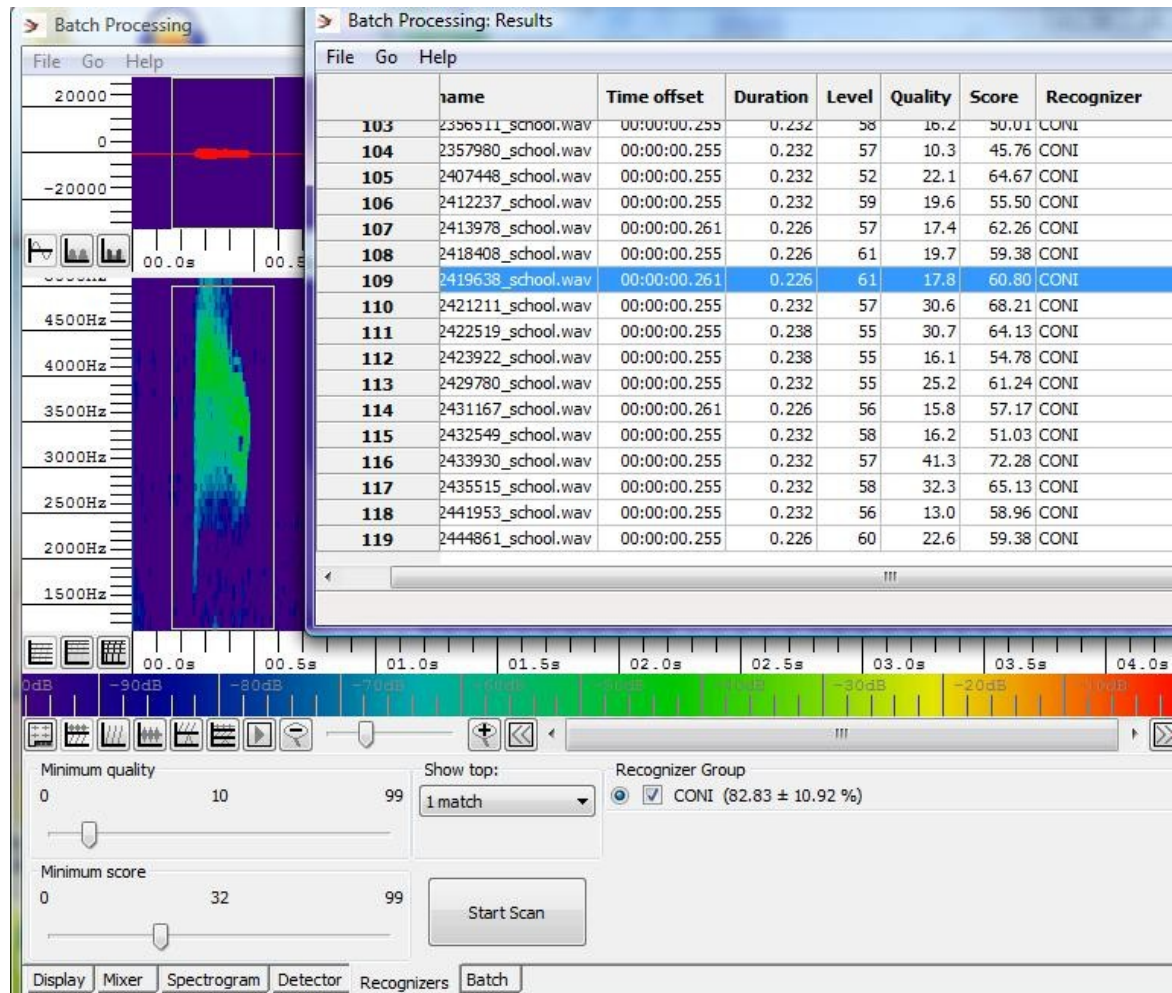


Figure 2-9: Batch processing snap-shot while common nighthawk, bird flight calls are being identified with Song Scope

Firstly, FFT is applied to the corresponding call. Then, during the preprocessing, Wiener and bandpass filters are applied to the weak signal to reduce the noise and



enhance the signal. Once the signal is cleaned, a log frequency transformation, similar to a mel scale, is performed. Before applying the signal detection algorithm, power normalization is performed on the signal to normalize the log power levels. For feature extraction, the features of interest are expressed as Discrete Cosine Transform coefficients and power level features. After the feature extraction, a classification algorithm, which is based on Hidden Markov Models, is applied. Two dimensional DCT and K-Means vector clustering are applied for automatic classification of the syllables and an HMM is built for each class in the database. Finally, additional statistical filters are applied to the classification algorithms to reduce the false positives.

A summary of Song Scope Algorithms is:

1. Preprocessing: Wiener filter  $\rightarrow$  bandpass filter  $\rightarrow$  log frequency scale transformation  $\rightarrow$  normalization of the log power levels
2. Signal Detection Techniques: First and the last vocalizations are detected by examining the total energy of the bandpass filter.
3. Feature Extraction: DCT coefficients and power level features are extracted.
4. Classification: HMM estimation with Viterbi training, DCT and K-Means Vector clustering algorithms are applied.
5. Additional Statistical Filters applied to improve classification algorithms

## **2.6 Bird Species Used in This Study**

In this work, nocturnal flight calls were recorded during the spring migration time. The species of birds that migrate in Northwest Ohio were researched and some of these are listed in Table 2.3 [62] [63].

Table 2.3: Spring migration table of Northwest Ohio

<b><u>Northwest Ohio Spring Migration Table</u></b>		
<b>EARLY MIGRANTS</b>	<b>MID-MIGRANTS</b>	<b>LATE-MIGRANTS</b>
<b>Dominant Species</b> -Ruby crowned Kinglet (male) -Hermit Thrush -Yellow rumped Warbler (male) -White throated Sparrow (male)  <b>Subdominant Species</b> -Nashville Warbler (male) -Western Palm Warbler -Black and white Warbler (male) -Song Sparrow -Swamp Sparrow  <b>Over Flight Species</b> -Yellow throated Warbler -Prairie Warbler -Worm eating Warbler -Louisiana Waterthrush -Kentucky Warbler -Hooded Warbler	<b>Dominant Species</b> -Blue Jay -Ruby crowned Kinglet (female) -Swainson's Thrush -Magnolia Warbler (male) -Yellow rumped Warbler (female) -White-throated Sparrow (female) - Savannah Sparrow  <b>Subdominant Species</b> -Veery -Tennessee Warbler -Nashville Warbler (female) -Yellow Warbler -Chestnut sided Warbler -Black and White Warbler (female) -Common Yellowthroat -Lincoln Sparrow	<b>Dominant Species</b> -Cedar Waxwing -Red eyed Vireo -Magnolia Warbler (female) -American Redstart -Indigo Bunting -Tennessee Warbler  <b>Subdominant Species</b> -Ruby throated Hummingbird -Warbling Vireo -Bay breasted Warbler -Wilson's Warbler -Canada Warbler

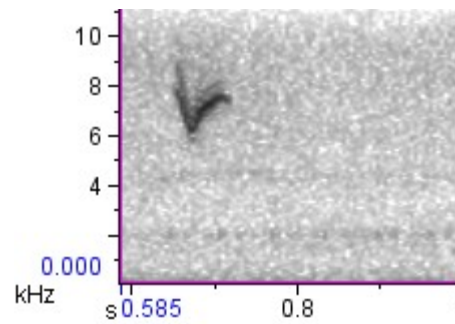
After preliminary examination of the recordings, flight calls of five species were studied in this work. Species were selected based on the number of their flight calls. The five Nocturnal flight calls that were analyzed in this work come from the following species: American Redstart (AMRE), Common Nighthawk (CONI), Savannah Sparrow (SAVS), Swainson's Thrush (SWTH) and Tennessee Warbler (TEWA). The names,

classes and scientific names of these birds are given in Table 2.4

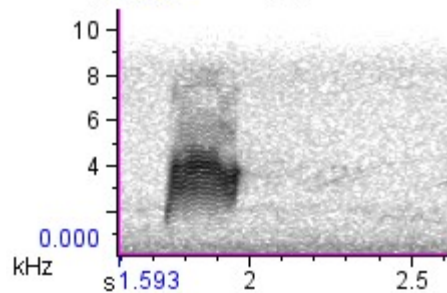
Table 2.4: Species, their classes and scientific names that are used in this work

Species	Class	Scientific Name
American Redstart	Warbler	<i>Setophaga ruticilla</i>
Common Nighthawk	Goatsuck	<i>Chordeiles minor</i>
Savannah Sparrow	Sparrow	<i>Passerculus sandwichensis</i>
Swainson's Thrush	Thrush	<i>Catharus ustulatus</i>
Tennessee Warbler	Warbler	<i>Oreothlypis peregrina</i>

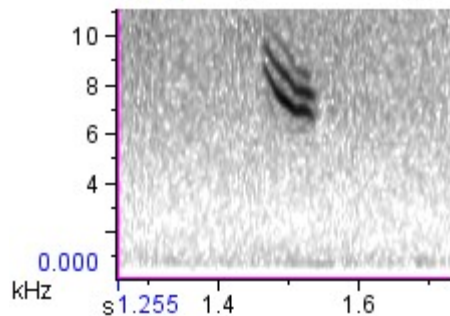
It should be noted that, although the common nighthawk is not a migrant, it was included in the study since many nocturnal flight calls were recorded. Figure 2-10 shows spectrograms and picture of the corresponding bird species. Spectrograms were obtained by using Raven Pro 1.3.



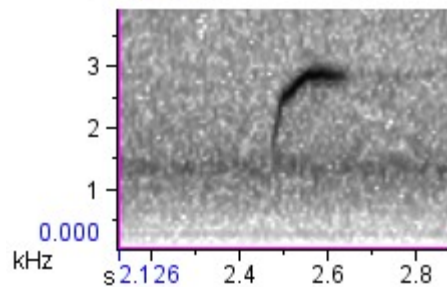
American  
Redstart



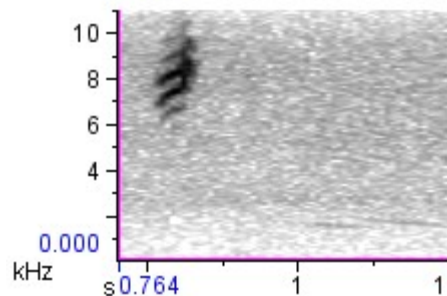
Common  
Nighthawk



Savannah  
Sparrow



Swainson's  
Thrush



Tennessee  
Warbler

Figure 2-10: Spectrograms and photographs, [www.allaboutbirds.com](http://www.allaboutbirds.com), of each species used in this work (obtained by Raven Pro 1.3)

In the following two chapters; feature extraction methods and classification techniques will be explained which were used for the identification of five species in Figure 2-10.

## Chapter 3

### Feature Extraction

Preprocessing, feature extraction and classifications are basic computational steps required in a bird flight call recognition system. The input of the flight call recognition system is bird flight calls. The raw flight call includes many features and non-relevant information such as background noise and characteristics of the recording device. These make raw input data complex and unsuitable as an input to a recognition system. Therefore, all non-relevant information should be removed during preprocessing [25]. In this project, preprocessing steps varies depending upon selected feature extraction method. Each species has flight calls of different characteristics and, therefore, after preprocessing, distinctive features of flight calls must be extracted in order to build a successful recognition system. Distinctive features drastically differ from species to species. Also, unique features should have low correlation with other features [64]. Extracted features should contain the maximum amount of information about the species within a smaller dimension. In another words, feature extraction takes the high dimensional data and transforms it to a low dimensional space [65]. Extracted features

are sent to a classifier which will specify class of unknown bird. Classification performance of classifiers highly depends on how well features represent the individual. If features of different individuals overlap, the misclassification percentage increases. In this thesis, discrete wavelet, mel frequency and image frequency based features are extracted. Feature extraction methods are described in following sections.

### **3.1 Wavelet Coefficients**

The wavelet transform represents the signal in a time-scale domain. Wavelet analysis gives a time-scale region. Whereas the Fourier Transform (FT) and Short-time Fourier Transform (STFT) give frequency and frequency-amplitude information respectively [66] [67]. While FT and STFT give constant resolutions at all frequencies, wavelet transform gives multiple resolutions. Fourier transform uses sinusoid wave, as a basis function to analyze the signal. However, wavelet transform uses wavelets as a basis function. The major advantage of wavelets is the ability to perform local analysis over a particular region of interest. This means that the width of the wavelet function changes for each of the frequency components. Wavelet transforms can be Continuous Wavelet Transform (CWT) and Discrete Wavelet Transform (DWT). The main difference between them is the scale of operation which can be either continuous or discrete.

#### **3.1.1 Continuous Wavelet Transform**

Wavelet transformation decomposes an input signal into its detail and approximation coefficients by using family functions. Family functions are generated by the scaling function ( $\phi$ ) and the wavelet function ( $\psi$ ). The scale function is called the father wavelet

and wavelet function is called the mother or basis function. Wavelet functions are formed by shifting (translation) and scaling (dilation or compression) the mother wavelet,  $\psi$ , as [66]:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (3.1)$$

where,  $a$  and  $b$  are real numbers. Parameter ' $a$ ' scales the wavelet function. Also, it dictates the time and frequency resolution of the wavelet transform. Parameter ' $b$ ' is a shifting parameter and it corresponds to the time information in the transform.

CWT performs an operation between the time domain signal and a basis function which is similar to convolution. CWT of a signal  $x(t)$ , with respect to a basis wavelet function is given as:

$$CWT(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \psi_{a,b}^*(t) dt \quad (3.2)$$

The original signal can be reconstructed from  $CWT(a, b)$  and the mother wavelet as shown in Equation 3.3. In order to reconstruct the original signal, the admissibility condition [68] has to be satisfied. The admissibility condition is given in Equation 3.4.

$$x(t) = \frac{1}{c_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{a^2} CWT(a, b) \psi_{a,b}(t) da. db \quad (3.3)$$

$$c_\psi = \int_0^{\infty} |\hat{\psi}(w)|^2 \frac{dw}{|w|} < \infty \quad (3.4)$$

In above equations,  $\hat{\psi}(w)$  is the Fourier transform of  $\psi(w)$  and  $c_\psi$  is the admissibility constant.



### 3.1.2 Discrete Wavelet Transform and Multi-Resolution Analysis

Unlike CWT, Discrete Wavelet Transform does not carry redundant information. DWT provides only a sufficient amount of information for both analysis and synthesis of the original signal leading to a decrease in computation time.

A discrete wavelet family can be obtained by making ‘a’ and ‘b’ discrete parameters as shown in Table 3.1 [66], [67].

Table 3.1 Equations for CWT and DWT

For a continuous time signal, $x(t)$	
Continuous wavelet analysis	Discrete wavelet analysis
$\psi_{a,b}(t) = \frac{1}{\sqrt{ a }} \psi\left(\frac{t-b}{a}\right)$	$\psi_{j,k}(t) = a_0^{-j/2} \psi(a_0^{-j}t - kb_0)$
$a \in R^+ - \{0\}, \quad b \in R$	$a = a_0^j, \quad b = k \cdot a_0^j \cdot b_0, \quad j, k \in Z$

In this thesis, discrete wavelet and scaling will be used. After substituting discrete parameters into Equation 3.1, discrete wavelet family ( $\psi_{j,k}$ ) is obtained as:

$$\psi_{j,k}(t) = a_0^{-j/2} \psi(a_0^{-j}t - kb_0) \quad (3.5)$$

Usually,  $a_0$  is selected as “2” and  $b_0$  is selected as “1” as in Equation 3.6. From this, discrete scale and positions would be obtained as powers of two. These selections will

make the analysis more efficient because computations in computer systems are based on dyadic scale.

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (3.6)$$

The analysis of the signal at different frequencies with different time scales is called Multi Resolution Analysis (MRA) [69]. Discrete wavelet transforms analyze the signal using MRA. The scaling function of MRA is defined as:

$$\phi_{j,k} = 2^{-j/2} \phi(2^j t - k) \quad (3.7)$$

A signal  $x(t)$  can be written as a series expansion in terms of the scaling function and wavelet function as [70]:

$$x(t) = \sum_{k=-\infty}^{\infty} c(j,k) \phi_{j,k}(t) + \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k) \psi_{j,k}(t) \quad (3.8)$$

where,  $c(j,k)$  and  $d(j,k)$  are the discrete scaling coefficients and the discrete wavelet coefficients, respectively. The first part of the summation in Equation 3.8 gives the approximation coefficients of the  $x(t)$  while the second part gives the detail coefficients.

The following equation is obtained by inserting the Equation 3.6 and Equation 3.7 into the Equation 3.8 [70]:

$$x(t) = \sum_{k=-\infty}^{\infty} c(j,k) 2^{-\frac{j}{2}} \phi(2^j t - k) + \sum_{k=-\infty}^{\infty} d(j,k) 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (3.9)$$

If the wavelet function is orthonormal to the scaling function, level  $j$  approximation coefficients and level  $j$  detail coefficients are obtained as seen in Equation 3.10 and Equation 3.11, respectively [70]:

$$c(j,k) = \int x(t) 2^{j/2} \phi(2^{-j}t - k) dt \quad (3.10)$$

$$d(j,k) = \int x(t) 2^{-\frac{j}{2}} \psi(2^{-j}t - k) dt \quad (3.11)$$

The DTW decomposes the input signal into its detail and approximation coefficients using the discrete scaling and the wavelet functions. From Equation 3.10 and Equation 3.11, it is seen that wavelet coefficients can be constructed by applying discrete low pass ( $H_0$ ) and high pass ( $G_0$ ) filters to the signal and then scaling it [70] [71] [25]. If the transform is wavelet decomposition, scaling is a down sampling by '2'. If the transform is a wavelet reconstruction, scaling is a up sampling by '2' [67]. The filtering procedure, at the basic level, is shown in Figure 3-1.

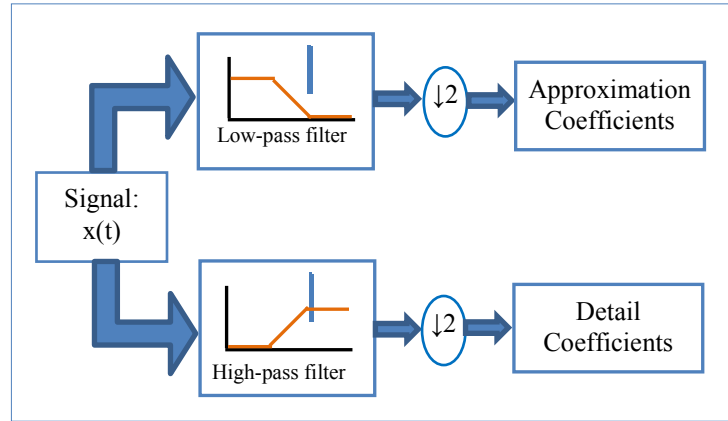


Figure 3-1: Filtering process of discrete wavelet transform

This practical algorithm was first developed by Mallat in 1988 and is called a two-channel subband coder [69]. It yields a fast wavelet transform which, when a signal is passed through it, yields wavelet coefficients at a much faster rate. The procedure of 3-level decomposition of signal using high pass and low pass filters is shown in Figure 3-2.

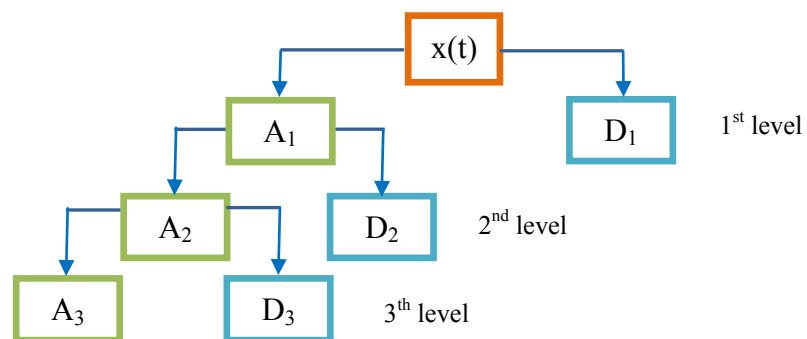


Figure 3-2: Three level discrete wavelet decomposition

### 3.1.3 Wavelet Families and Features

There are numerous types of wavelet families, such as Daubechies (db), Coiflets (coif), Symlets (sym), Discrete Meyer (dmey) and Biorthogonal (bior) [25] [26]. Choosing the wavelet function and decomposition level is critical in wavelet transformation when it comes to extracting the most relevant information from the signals. With that said, there is currently no consensus on which particular wavelet family works most optimally to remove a particular type of noise. As an example, properties of three different wavelet families will be presented in this subsection: Morlet, Daubechis and Symlets. The Morlet wavelet was formulated by J. Morlet for a study of seismic data [72] [73]. The Morlet wavelet function is show in Figure 3-3.

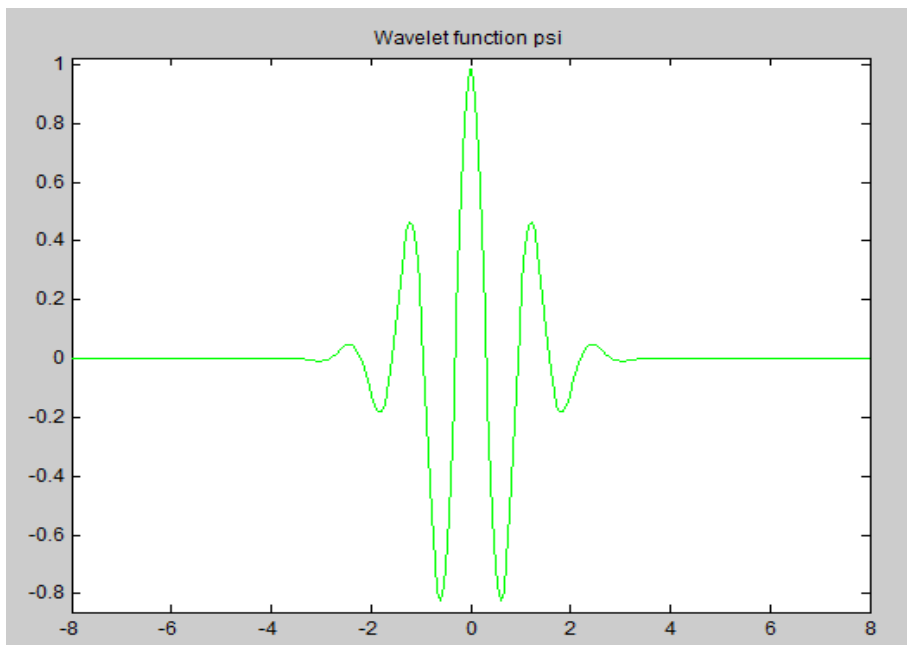


Figure 3-3: Morlet wavelet function

Daubechies and Symlet wavelets are orthogonal wavelets that have the highest number of vanishing moments for a given support width. In the naming convention, db-i or sym-i, i is an integer that denotes the order, e.g. db10 is 10<sup>th</sup> order Daubechies wavelet and sym8 is 8<sup>th</sup> order Symlets wavelets. An i<sup>th</sup> order wavelet has i vanishing moments, a support width of 2i-1 and filter length of 2i. These wavelets are suitable for use with both the continuous wavelet transform and the discrete wavelet transform. The difference between these two wavelet functions is that Daubechies wavelets are far from symmetry while Symlet wavelets are nearly symmetric. The Daubechies wavelet is a common choice for the analysis and synthesis filters, because it possesses several nice properties. The filter produced by the Daubechies family of wavelets are orthogonal and the frequency response has maximum flatness at  $\omega = 0$  and  $\omega = \pi$ . The second property leads to excellent results when Daubechies filters are used for the DWT decomposition and reconstruction of a large class of signals. The Daubechies filters are compactly supported i.e the impulse response is zero outside a certain time interval. As an example, Figure 3-4 shows wavelet ( $\psi$ ) and scaling functions ( $\Phi$ ) for fourth order Symlet and Daubechies wavelets.

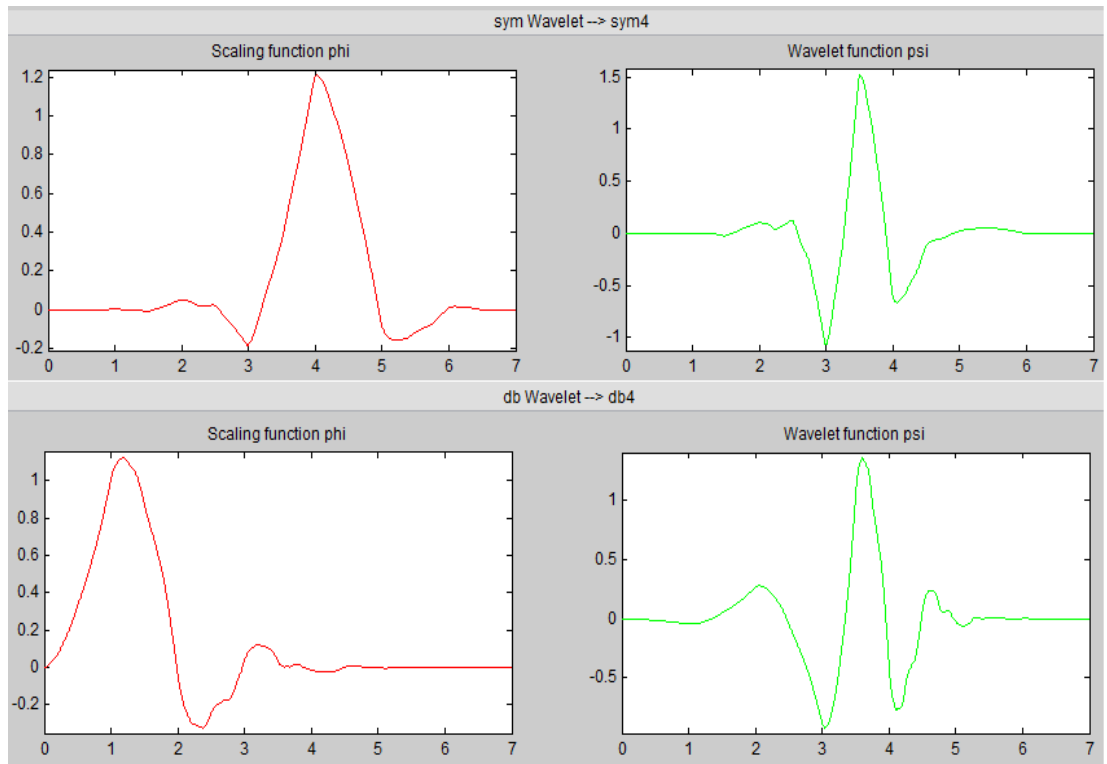


Figure 3-4: Scaling and wavelet functions of db 10 and sym 10 wavelet families

Daubechis wavelet families have been shown to be practical in speech recognition [26]. In this thesis, db10 wavelet families were used with 6-level decomposition.

Once the approximation and detail coefficients were extracted, basic statistics were applied to detail coefficients in order to generate feature vectors. For each bird call, fifty-four feature vectors were obtained. Feature vectors carry the information of the range, maximum, minimum, mode, median, mean, mean absolute deviation, median absolute deviation and standard deviation of the detailed coefficients.

## 3.2 Wavelet-denoising Based Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficient's (MFCC's) are obtained from the mel frequency cepstral spectrum which is very similar to the perceptual perspective of the human auditory system [74]. Human ears hear frequencies lower than 1000 Hz with a linear scale and the frequencies higher than 1000 Hz with a logarithmic scale. MFCCs mimic this nonlinear loudness perception [74].

Traditional preprocessing of the MFCC consists of filtering and normalization. In this thesis, wavelet-denoising has been applied to the signals for noise reduction. With wavelet-denoising, more adequate features have been obtained for the flight call recognition system. A general block diagram of the system is given in Figure 3-5.

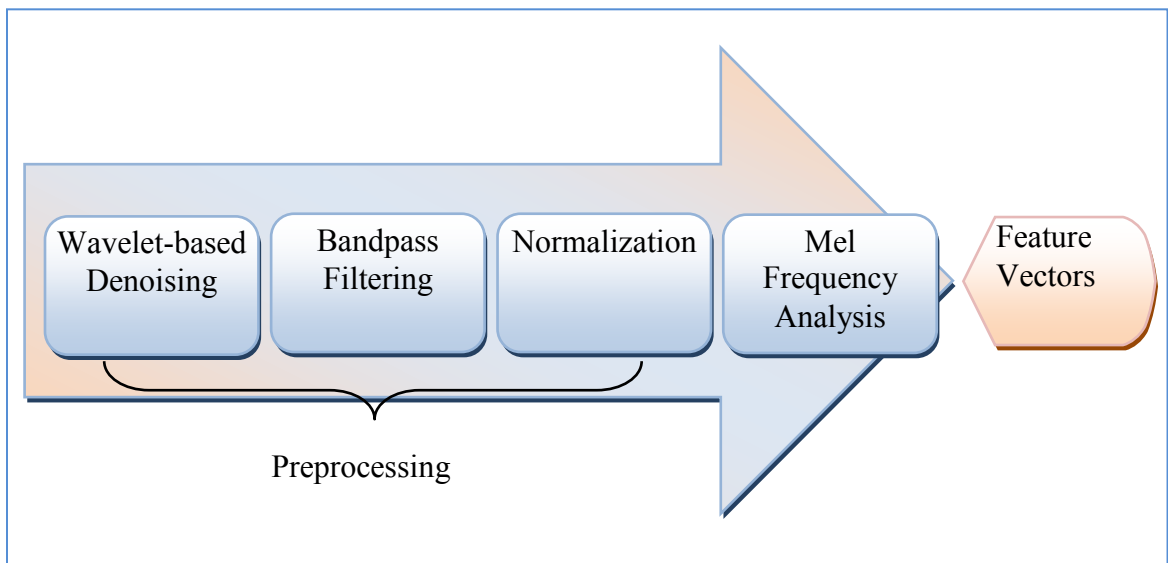


Figure 3-5: Wavelet denoising based MFCC feature extraction scheme

### 3.2.1 Wavelet-denoising

Discrete Wavelet Transformation (DWT) was explained in Section 3.1.2. In DWT the audio signal is passed through complementary filters whose output consists of a low-



frequency component and a high- frequency component. They are popularly named as approximations and details. A single level decomposition gives two components of coefficients namely approximation and detail. The decomposition can be applied to multiple stages. With each level of decomposition, the output of the low pass filter contains more of the original signal content which is called approximation coefficients. Approximation coefficients are free from high frequency components which are nothing but noise in the signal. Therefore DWT can be used to denoise the signal. Decomposition of the signal is explained as following:

Initially the signal  $x(t)$  is decomposed to one level and the corresponding approximations and details are obtained. The decomposition of a signal  $x(t)$  is given by:

$$x(t) = cA + cD \quad (3.14)$$

where,  $cA$  is an approximation coefficient and  $cD$  is a detail coefficient. It is observed that, with decomposition, low-frequency components, that are essentially actual signals, are separated from the corresponding high-frequency components. With multilevel decomposition, still more high-frequency components are separated. The signal  $x(t)$  is then subjected to multilevel decomposition with levels 3, 4, 5, 6, etc. and the corresponding approximations and details are analyzed. Multilevel (n-level) decomposition of  $x(t)$  is given by,

$$x(t) = cAn + cDn + cD(n - 1) + cD(n - 2) \dots + cD1 \quad (3.15)$$

where,  $cA_n$  is  $n$ -level approximation and  $cD_n, cD(n-1), \dots, cD1$  are corresponding detail coefficients.

Most of the high frequency components are separated from the signal in the first detail  $cD1$  itself. Details obtained in the consecutive levels, further separate the noise components from the signal thus leaving the signal free from noise. The signal which has been reduced in the number of samples can be up sampled or interpolated to get back the original length. The approximation which is noise free is now suitable for analysis.

It is observed that the detail  $cD1$  consists of the high frequency components of the signal. This detail values can be manipulated by limiting the amplitudes of the noise. Marginalization of the detail should be done carefully as this detail may consist of high frequency components which are an integral part of the actual signal. The threshold value specifies the levels of noise that are to be removed. Therefore, choosing the threshold value is an important aspect for denoising without the loss of any useful data.

An example of four-level wavelet-based denoising technique is explained in the following figures. Figure 3-6 is time-domain representation of a real signal, a Swainson's thrush flight call. Spectral responses before denoising are given in Figure 3-7. It can be seen that most of the signal is distributed in the lower frequency regions and glitches (noise) are uniformly distributed throughout the spectrum.

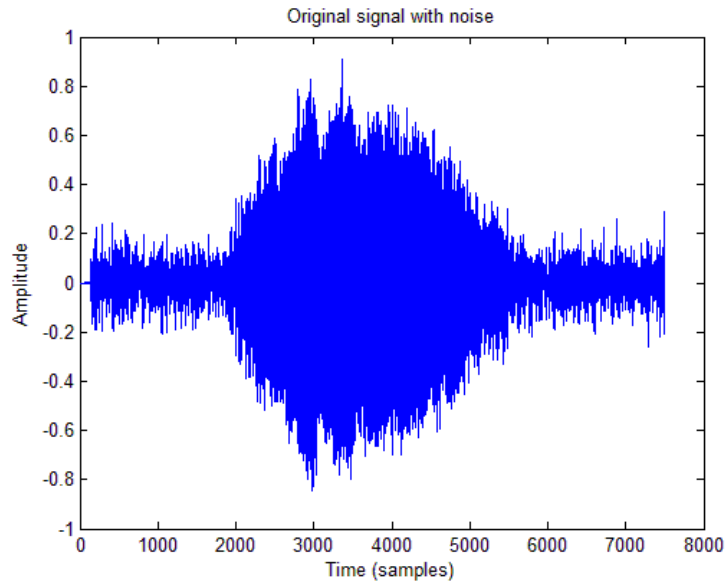


Figure 3-6: Original signal, Swainson's Thrush call, with noise

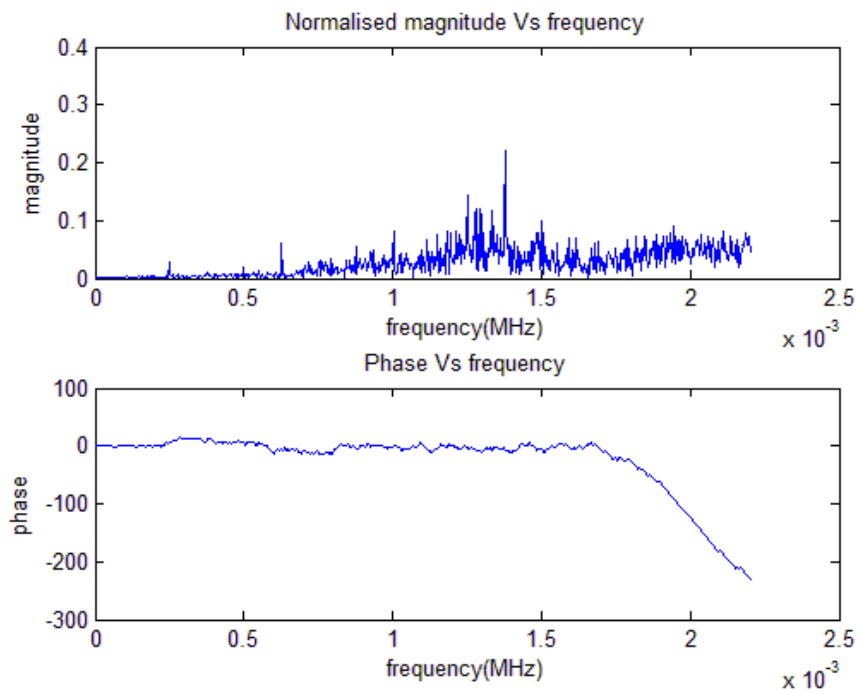


Figure 3-7: Spectral responses of the real signal

The signal is then decomposed into low-frequency (approximations) and high-frequency (details) signals. Figure 3-8 and Figure 3-9 show the detail and approximation coefficients after decomposition, respectively.

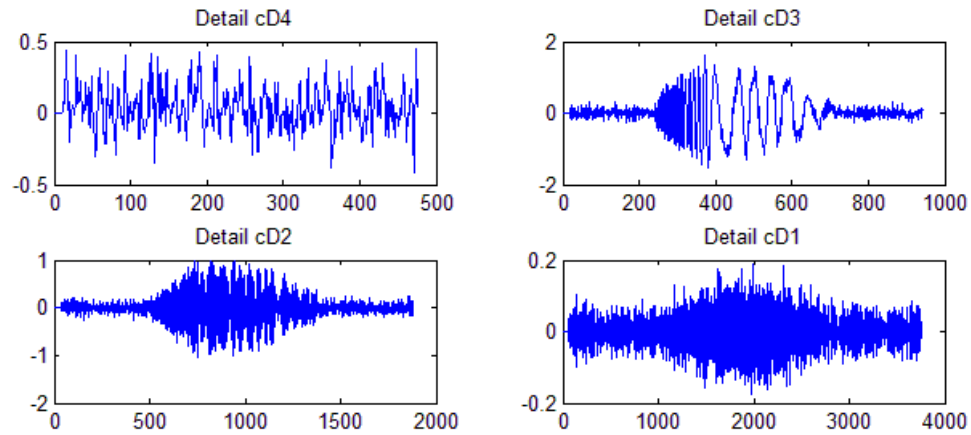


Figure 3-8: Detail coefficients of the signal.

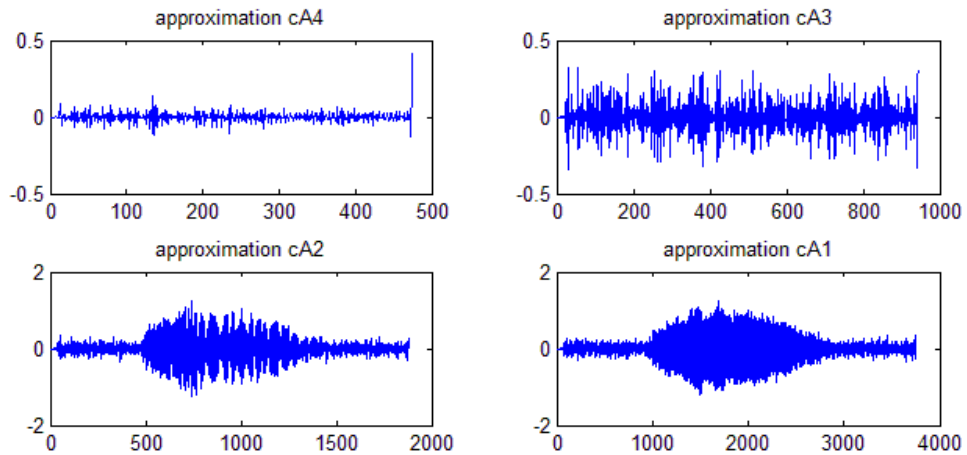


Figure 3-9: Approximation coefficients of the signal

The information of interest normally resides in the low frequency components [25] [68]. At the 4<sup>th</sup> level, approximations will have signal and noise, can be taken directly and are free from high-frequency components. Figure 3-10 shows the real signal and the denoised signal after the wavelet decomposition.

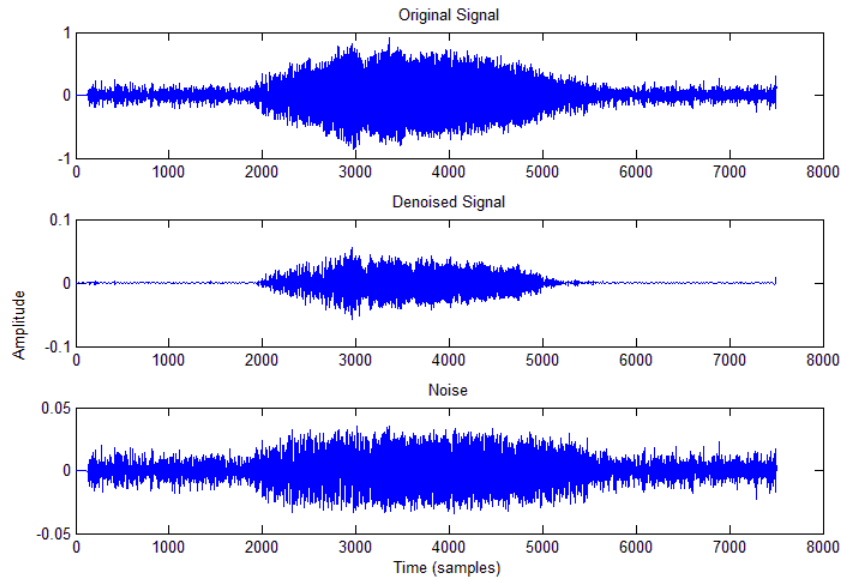


Figure 3-10: Real signal, noise and denoised signal after wavelet denoising

A spectrum plot can be obtained again after denoising as seen in Figure 3-11. Compared to Figure 3-7, it is observed that there is less area under the high frequency region, confirming that there has been a considerable removal of noise.

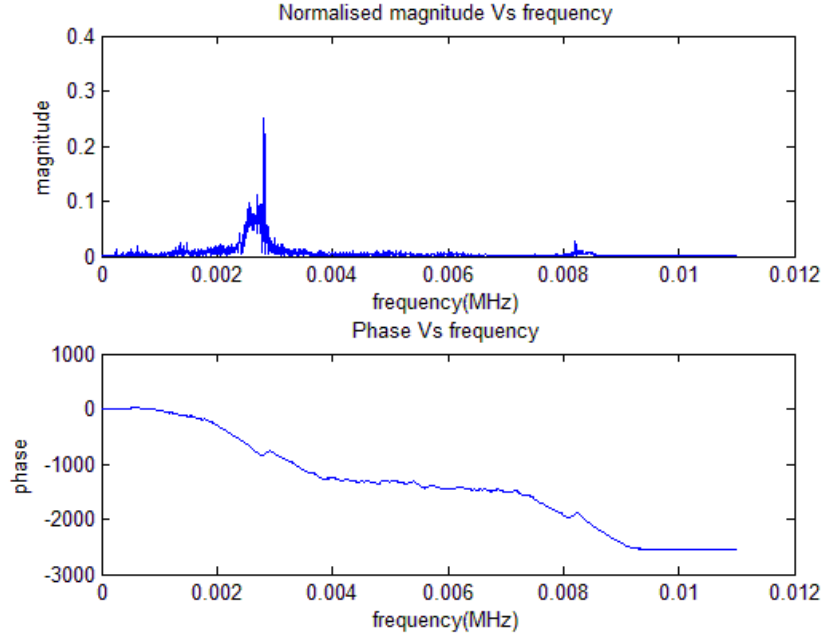


Figure 3-11: Spectral responses after wavelet-based denoising

Based on preliminary tests, db4 and Simlet wavelet families yielded good results for flight call signals. The number of decomposition levels depends on the application and the satisfaction of the user. With 3-4 levels of decomposition, flight call signals will be denoised to an extent that is sufficient for this project. If an integral part of the signal has high frequency components, more levels of decomposition should be applied in order to extract the original signal from noise.

### 3.2.2 Filtering and Normalization

After wavelet-based denoising, all signals are filtered with bandpass filters. Bandpass filters clear the information below 1300 Hz and above 11000 Hz. These frequency ranges were chosen because the lowest frequency component of the flight calls is observed as 1300 Hz and the highest frequency component is observed as 11000 Hz. Cleaned signals

have been normalized due to possible mismatches between training and test conditions. The amount of variation in the data is reduced by normalization. During normalization, every sample of the bird flight call is divided by the highest amplitude value. The mean of the flight call signals are then subtracted from the normalized signal to remove the unwanted DC offset.

### **3.2.3 Mel Frequency Cepstral Coefficients**

Mel Frequency Cepstral Coefficients (MFCC) are short term power spectral features of a sound [74]. To obtain the mel frequency features, the signal of interest is first divided into frames. Then each frame is preprocessed by applying a windowing function to reduce the discontinuities at the beginning and end of the signal. The next step is to take the fast Fourier transform of each windowed frame and convert frames into frequency domain. Then, the frequency spectrum of the signal is multiplied by the mel frequency filter bank [75]. The mel frequency filter bank consists of triangular filters that have a triangular bandpass frequency response. With the filter bank defined, the next step is to find the coefficients of the first frame by calculating the discrete Cosine Transform. Figure 3-12 gives a block diagram illustrating how MFCC features were obtained.

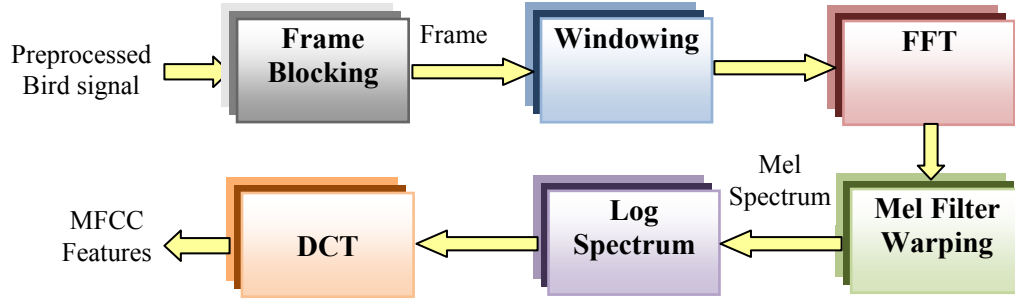


Figure 3-1 2: Block diagram for MFCC feature extraction method

In this work, equations similar to those found in [75] were used to derive the MATLAB program. Initially, the mel spectrum was described. Unlike cepstral spectrums which have unequal frequency spacing; the mel scale cepstral spectrums have equal (linear) spacing if the frequency is higher than 1 kHz and unequal (logarithmic) spacing otherwise. Mel frequency is defined as:

$$Mel(f) = 2595 * \log_{10}(1 + \frac{f}{700}) \quad (3.16)$$

The program starts by frame blocking the bird call with each frame having the length of 512, and applying a Hamming Window to the frame. The Hamming Window can be mathematically represented as:

$$w(n) = 0.54 - 0.46\cos(\frac{\pi n}{M}) \quad (3.17)$$

where  $n = 1, 2, 3, \dots, 512$  and  $M$  (frame length) is 512. Then, to each window frame, a

Discrete Fourier Transform (DFT) is applied as:

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n) \exp\left(-j2\pi kn/N\right) \quad (3.18)$$



where  $k = 0, 1, \dots, N - 1$  represents the frequency and, with birds reaching almost 11 kHz in their calls,  $N$  is set as 11,000 and again  $n = 1, 2, 3, \dots, 512$ . After Fourier coefficients are obtained, the magnitude spectrum of  $X(k)$  is taken as in Equation 3.19, to reject the imaginary parts.

$$|X(k)| = \sum_{n=0}^{n-1} \sqrt{\text{real}(X(k))^2 + \text{imaginary}(X(k))^2} \quad (3.19)$$

With the magnitude of the DFT derived, the next step is to scale the magnitude spectrum logarithmically. This is performed using Equation 3.20, where  $m = 1, 2, \dots, M$ ,  $M$  is the number of filter banks and  $H(k, m)$  is the filter bank. The mel filter bank, which consists of triangular filters, is defined in Equation 3.21 where  $f_c(m)$  is the center frequencies.

$$X'(m) = \ln(\sum_{k=0}^{N-1} |X(k)| \cdot H(k, m)) \quad (3.20)$$

$$H(k, m) = \begin{cases} 0 & \text{for } f(k) < f_c(m-1) \\ \frac{f(k) - f_c(m-1)}{f_c(m) - f_c(m-1)} & \text{for } f_c(m-1) \leq f(k) < f_c(m) \\ \frac{f_c(m) - f(k)}{f_c(m) - f_c(m+1)} & \text{for } f_c(m) \leq f(k) < f_c(m+1) \\ 0 & \text{for } f(k) \geq f_c(m+1) \end{cases} \quad (3.21)$$

In this work, twenty four triangular bandpass filters were used to construct the mel filter bank. Figure 3-13 shows the MATLAB plot of a constructed mel spaced filter bank. The bandwidth and spacing of the each filter are in accord with the mel scale in the frequency domain.

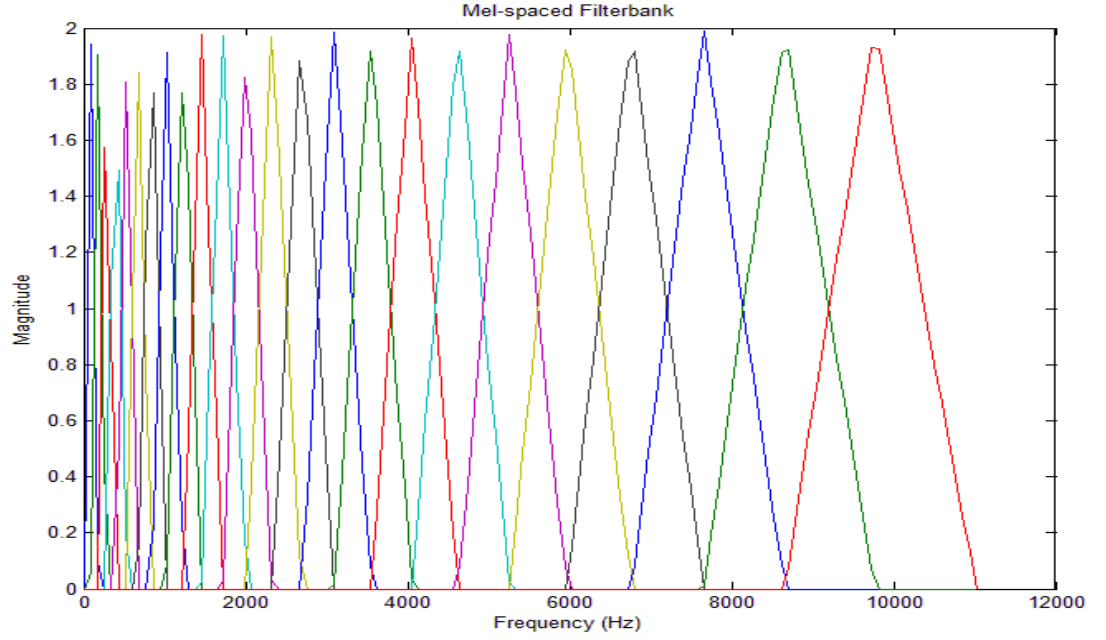


Figure 3-13: Mel spaced filterbank with 24 triangular bandpass filters

Equation 3.22 is used to derive the mel scale, which converts Hertz into Mel.  $f$  is set as **11kHz**. The corresponding mel scale is obtained as in Figure 3-14

$$\phi(k) = 2595 \log_{10} \left( \frac{f}{700} + 1 \right) \quad (3.22)$$

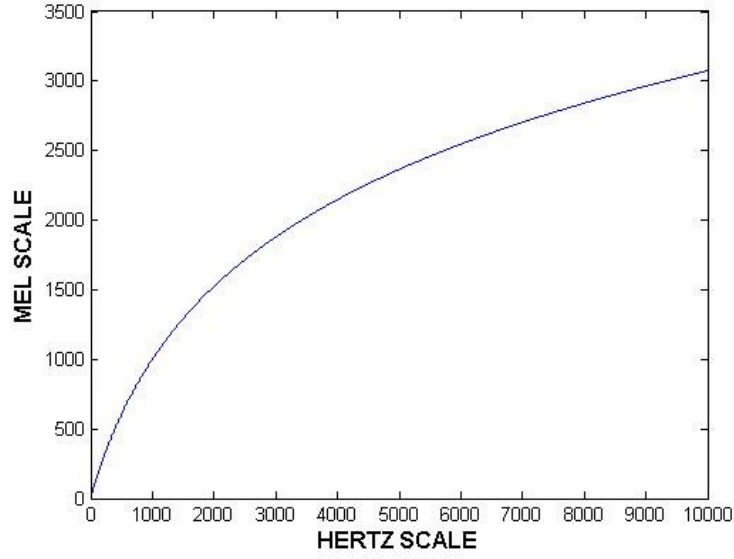


Figure 3-14: Mel scale vs. Hertz scale

Once the Mel Scale is derived, it is solved for a fixed frequency resolution using Equation 3.23; where  $\phi_{max}$  is the highest frequency and  $\phi_{min}$  is the lowest frequency of the filter bank on mel scale.

$$\Delta\phi = (\phi_{max} - \phi_{min}) / (M + 1) \quad (3.23)$$

In the algorithm; with a maximum frequency of 11 kHz,  $\phi_{max}$  is set as **17.5 kHz** and  $\phi_{min}$  is set as **0 Hz** using Equation 3.22. The center frequencies of the mel scale,  $f_c(m)$ , are calculated using the inverse of Equation 3.22, which is expressed as:

$$f_c(m) = 700 \left( 10^{\frac{\phi_c(m)}{2595}} - 1 \right) \quad (3.24)$$

Then, by substituting the obtained center frequencies and the frequency function into Equation 3.21; the mel filter bank is obtained. With the filter bank defined, the next step is to find the coefficients of the first frame. The first step is converting from the Hertz frequency scale into the mel frequency scale using Equation 3.22. Then the Discrete Cosine Transform (DCT) of the created MFCC's are found using:

$$c(l) = \sum_{m=1}^M X'(m) \cos\left(l \frac{\pi}{M} \left(m - \frac{1}{2}\right)\right) \quad (3.25)$$

Finally, from the DCT, the coefficients are placed in descending order of information. The higher order coefficients are discarded in order to remove the components due to the periodic excitation source. The first DCT coefficient is also discarded because it represents the average power of the call and skews the results of the MFCC's harmonics. In this work, the next twelve coefficients were taken, of the twenty four, and the same steps were applied to the next frame of the bird call. Thus, for each frame; a feature vector was calculated that consisted of twelve coefficients. Then, averaged MFCC's were obtained by overall frames. This yielded twelve features for each bird call.

### 3.3 Spectrogram-based Image Frequency Statistics (SIFS)

Fourier theory indicates that a signal can be represented as a sum of an infinite series of sine and cosine [43]. Fast Fourier Transform (FFT) [39] can be used as a feature extraction technique. The drawback of FFT is its lack of time-localization analysis. Therefore, although FFT is appropriate for stationary signals, it is not well-suited for non-stationary signals. This problem was solved by Dennis Gabor [76], when he first developed Short Time Fourier Transform (STFT) technique. The STFT partitions the

non-stationary signal into windows of signals with short periods of time. Then, those windows are treated as stationary signals. FFT is applied to each window separately.

Bird calls used in this work were not distinguishable in waveforms. Therefore, measurements for feature extraction can be achieved through use of spectrograms. Example of a waveform and a spectrogram for a Swainson's thrush call is given in Figure 3-15.

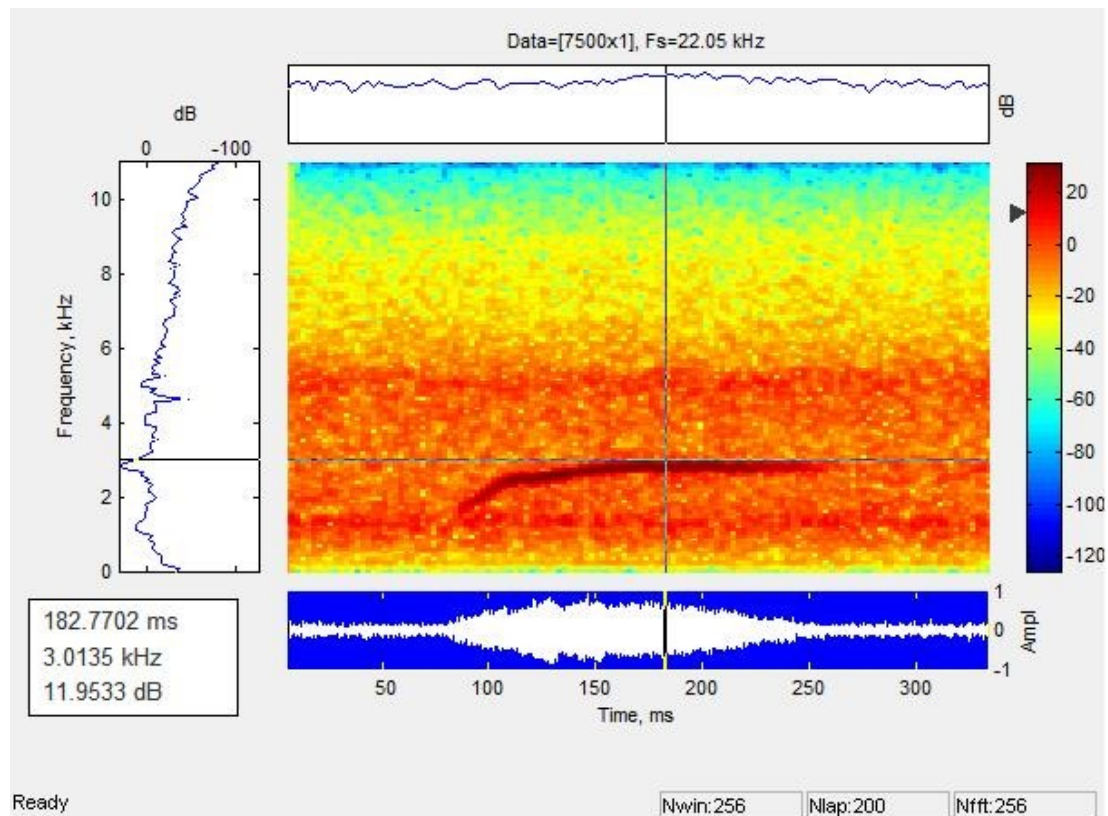


Figure 3-15: Waveform and spectrogram of Swainson's Thrush call

A spectrogram is constructed by applying STFT to the signal [77]. Discrete time STFT is defined as [78]:

$$STFT(m, f) = \int_{-\infty}^{\infty} x(n)w(n - m)e^{-j2\pi ft} \quad (3.26)$$

Where,  $x(n)$  is the signal,  $w(n)$  is the window function and  $m$  is the index of the signal.

Bird calls can be represented with Equation 3.26, which can be visualized as an image called spectrogram. A spectrogram shows how spectral characteristics of the signal vary with time. A spectrogram is constructed by applying the FFTs vertically in an image, and dividing a different column for each data segment in the time domain. Conventionally, the vertical axis represents the frequency information and horizontal axis represents the time information. At a certain frequency and time, the magnitude of the value is proportional with the spectrogram, which is considered as a third dimension. Amplitude is indicated by a variety of colors. Darker pixels indicate peaks in the spectrogram. A 3D spectrogram of a Swainson's Thrush test call is shown in Figure 3-16.

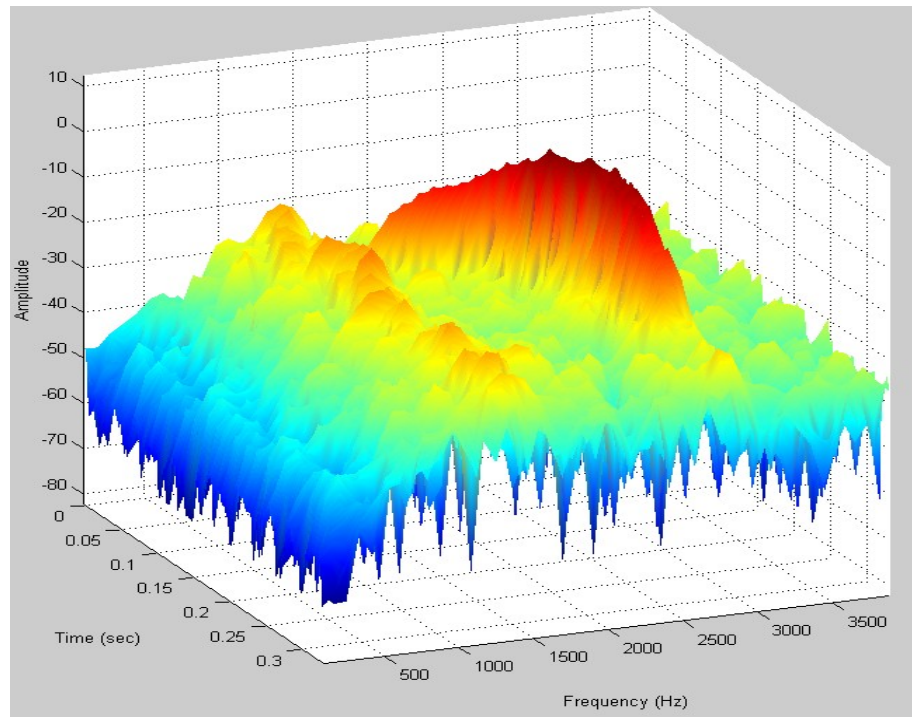


Figure 3-16: 3D spectrogram of Swainson's Thrush test call.

The technique of Spectrogram-based Image Frequency Statistics (SIFT) feature extraction is explained with an example using three different bird calls as shown in Figure 3-17. Spectrograms of thrush, sparrow and warbler real calls are shown and it can be observed that these bird calls do not have the same length, amplitude, or noise distribution.

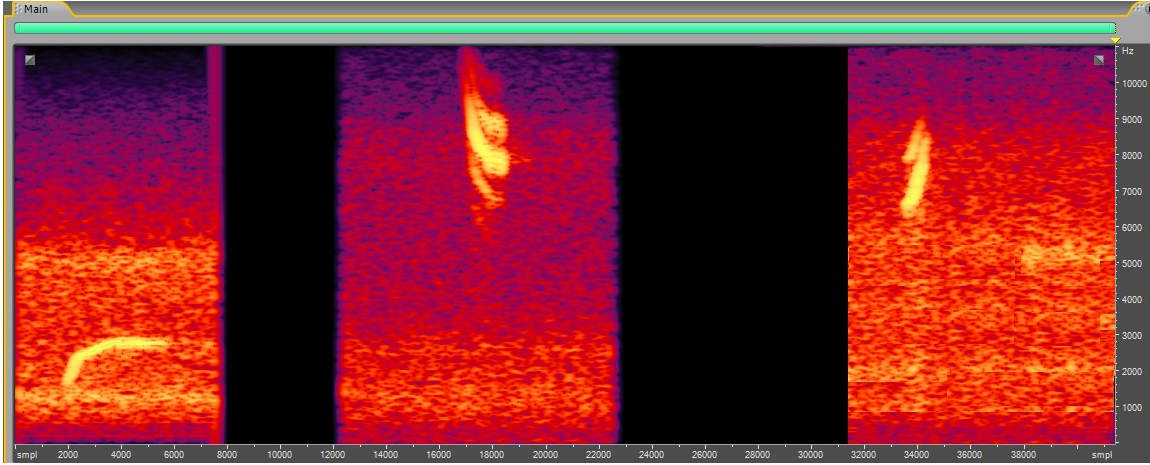


Figure 3-17: Spectrograms of Swainson's Thrush, Savannah Sparrow and Tennessee Warbler.

In order to extract Spectrogram-based Image Frequency Statistics (SIFS) features, several observations are made from Figure 3.17:

- Bird calls had the highest amplitude in the signal.
- Bird calls were most obviously distinguishable in the time-frequency domain.
- Low frequencies had the highest noise amplitude noise.

From above observations, SIFS features were extracted using following operations:

1. The spectrogram of each call was extracted by setting the sampling frequency to 22050 Hz, window length to 512, overlap to 50% and number of FFT points to 256.
2. The lowest frequency of a bird call in the database was 1300 Hz. Therefore, bottom most frequency responses of the spectrogram were filtered. Bottom most frequency responses were obtained as 30 using the following relation:

$$Nc = (Fc) \frac{2}{Fs} (NFFT) \quad (3.27)$$



where,  $N_c$  is the bottom most frequency responses,  $F_s$  is the sampling frequency,  $NFFT$  is number of FFT points and  $F_c$  is the lowest call frequency.

3. The position of spectral responses inside the spectrogram was of more interest than their amplitude. Therefore, all spectral responses of amplitude at least 10% from the amplitude of the overall maximum spectral peak of the spectrogram were set to 1. Other spectral responses were set to 0. 10% as a custom threshold that keeps as much of the signal as possible while still discarding a reasonable amount of noise. A threshold value was chosen after making some observations for each species' spectrograms. The resulting image was a binary image.

4. A dilation operation was performed on binary images to enhance the continuity in the call signature by reducing holes and gaps between objects and expanding the white pixel regions. Images were dilated with a structuring element [79]:

$$D = A \oplus B \quad (3.28)$$

where, A is an image, B is a structuring element and  $\oplus$  is a dilation operator. Dilation is a type of morphological operation [80] that enlarges or flattens objects in an image [80]. When the dilation process is applied to binary images, it is very similar to a convolution process. The structuring element is the most important component of a dilation process. In fact the structuring element can be thought of as a small binary image with user defined size and shape. It can be a shape of square, rectangle, circle or a line.

The structuring element determines how objects in an image will be flattened. The origin of the structuring element,  $B$ , is laid over each pixel of the input image. The process of dilation can be summarized in two main steps [81] [82]:

- a. If the origin of  $B$  encounters with a zero-valued pixel, nothing happens and operation continues with the next pixel.
- b. If the origin of  $B$  coincides with a one-valued pixel; pixels of  $B$  and pixels of the image that are covered by  $B$  are subjected to a logical “OR” operation.

If the purpose of a dilation operation is to soften sharp lines, a circular structuring element is used. If the purpose is to enlarge the width and height of an object at the same proportion, a square structuring element is usually used. In this thesis, the object of the structuring element was to enlarge the area of chirp spectrogram. Chirps were observed as small areas or lines and, therefore, the structural element was created as a vertical line of 5 pixels. An empirical threshold, based on the observation that flight calls were visible, was at least 4 consecutive windows. The dilation then enlarged the area of the chirp spectrogram to at least 20 pixels. The area and bounding box of each region was computed and all regions of 8-connected white pixels were mapped. If the area of the region was lower than 20 pixels, the region was blackened and ignored.

**5.** The largest signature of a flight call was observed as 50 columns. Therefore, the image with an exact width of 50 columns was obtained by removing black columns and stretching the resulting image, as shown in Figure 3-18. New features which represent thrush, sparrow and warbler flight calls can be seen in Figure 3-18.



Figure 3-18: Features after cleaned spectrogram

6. The last step of the SIFS technique is dimensional reduction. Features that were obtained previously are computationally inefficient. Therefore, the number of features that represent the flight calls needs to be reduced. Thus, frequency statistics were applied for feature reduction. Frequency statistics compute the lowest, highest, mean and median frequencies, assuming that the input images are cleaned spectrograms. Properties needed are as follows:

- Bounding box - the smallest rectangle that covers all white pixels
- Centroid - the centroid of the white pixels
- Pixel list - a list of the coordinates of all pixels

All the properties were obtained by using “regionprop function” in MATLAB.

Features of lowest, highest, mean and median frequencies were computed for the whole image, first  $3/7^{\text{th}}$ , middle  $3/7^{\text{th}}$  and last  $3/7^{\text{th}}$  of the image as shown in Figure 3-19.

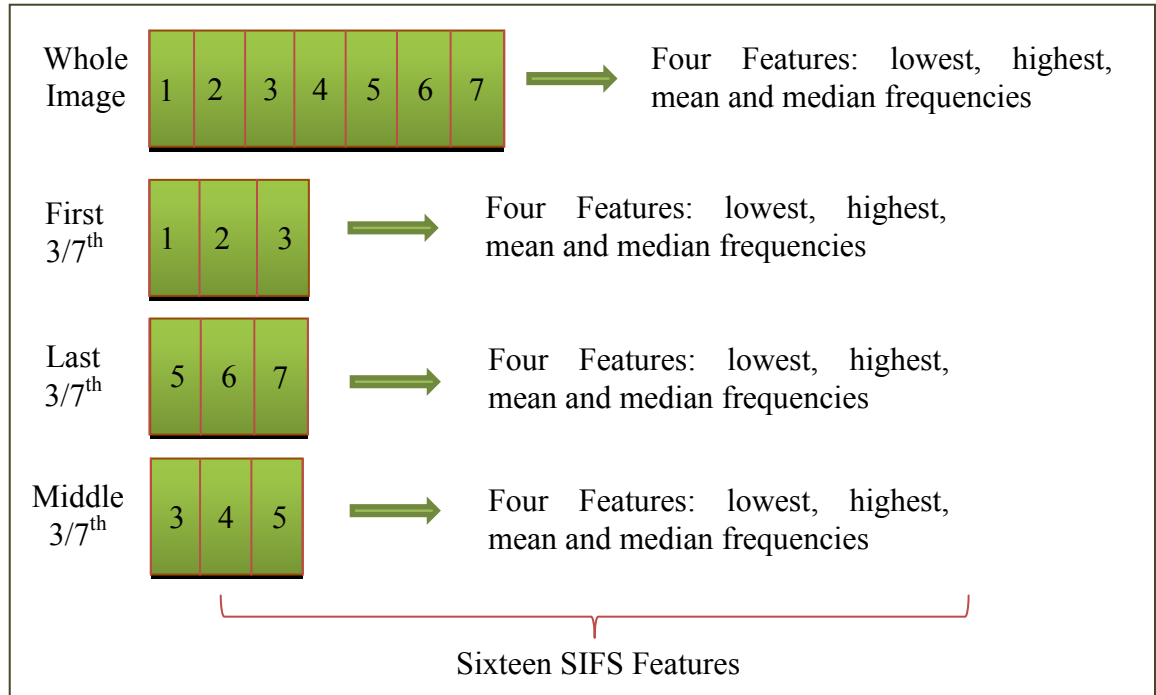


Figure 3-19: Dimension reduction part of the SIFS feature extraction method

Feature vectors of a flight call became more distinctive by extracting statistical features for different parts of the image. Thus, each call provided sixteen features and they could be used for the classification. The SIFS feature extraction scheme is summarized in Figure 3-20. These steps were applied to all test and training calls.

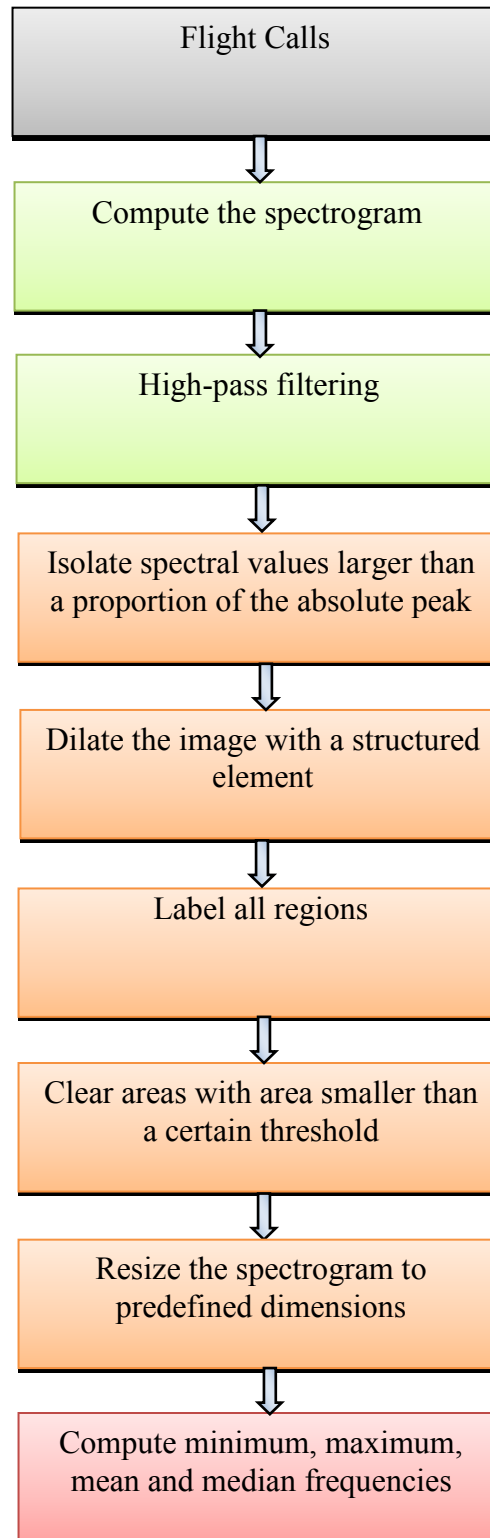


Figure 3-20: Summary of the SIFS Technique

### 3.4 Mixed MFCC and SIFS (MMS)

Features from different methods can be hybridized in order to achieve higher performance than features that are obtained from one method. MMS features were created by combining MFCC and SIFS features as shown in Figure 3-21. Mixed features carry the Mel Frequency information of a call, as well as frequency statistics of its image, which are obtained from the spectrogram. MFCC features form the first half of the feature vector while SIFS features form the second half of the feature vector resulting in 28 features for each call. Assume a nocturnal flight call has twelve MFCC features and sixteen SIFS features that are in the form of:

$$MFCC_{features} = \{MFCC_1, MFCC_2, \dots MFCC_{12}\} \quad (3.29)$$

$$SIFS_{features} = \{SIFS_1, SIFS_2, \dots SIFS_{16}\} \quad (3.30)$$

Then the hybridized feature vector can be represented as:

$$\left\{ \begin{matrix} MMS \\ Features \end{matrix} \right\} = \{MFCC_1, MFCC_2 \dots MFCC_{12}, SIFS_1, SIFS_2, \dots SIFS_{16},\} \quad (3.31)$$

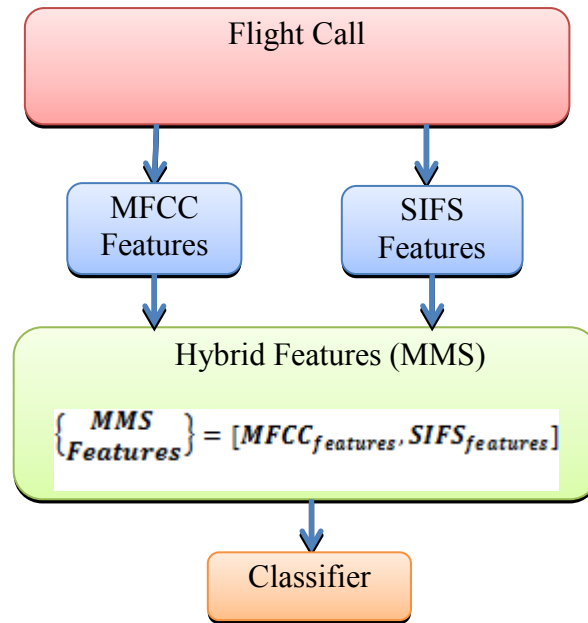


Figure 3-21: Recognition block diagram with MMS feature extraction

In this chapter, four different feature extraction methods were implemented to obtain unique feature vectors for each bird flight call. Each of these vectors will be fed into different classifiers which will be explained in the next chapter for species recognition.

# Chapter 4

## Classification

Extracted features are classified using machine learning techniques. There are different classification techniques in the context of machine learning. The aim of classification techniques is to classify unknown objects based on the classification measurements [83]. In this thesis, four different types of classifiers were implemented to classify bird flight calls. To this end, k-nearest neighbor (k-NN), Hidden Markov Models (HMM), Multilayer Perceptron (MLP) and Evolutionary Neural Network (ENN) are used.

### 4.1 K-nearest Neighbor Classifier (k-NN)

k-NN is a supervised classifier, and, although it is simple, it is very powerful when classifying data [83]. k-NN classifiers classify test samples based on the similarities between training and test samples in feature space, where similarity is measured in distance metric. Distance metrics include Euclidean, Mahalanobis, Correlation and Cosine. The number of nearest neighbors that must be checked to assign the unlabeled test data is referred to as k. The optimum distance metric and number of neighbors must



be determined for satisfactory classification results. The algorithm starts by labeling the training data. Parameters of the algorithm are defined by setting the value of k and the type of similarity measure. According to the similarity measure, the k closest points (nearest neighbors) are found for each test data. Then, each test data is assigned to a class, whose label appears as the majority in the k nearest neighbors.

k-NN is a supervised learning algorithm therefore, the class of each training data ( $Y = \{y_1, y_2, y_3, \dots, y_n\}$ ), is known initially. If the test data is assumed to have n members ( $X = \{x_1, x_2, x_3, \dots, x_n\}$ ); then steps required to implement the k-NN algorithm are explained as follows:

1. Define the number of neighbors, k.
2. Calculate distances between the query instance test data,  $x_1$ , and all the training

samples ( $\{y_1, y_2, y_3, \dots, y_n\}$ ). From this, a distance vector is obtained as:

$$D[d_1(x_1) \ d_2(x_1) \ d_3(x_1) \ \dots \ d_n(x_1)] = Dist(x_1, y_i)_{i=1}^n \quad (4.1)$$

3. Sort distance values in a distance vector in ascending order as:

$$d_1(x_1) \leq d_2(x_1) \leq \dots \leq d_n(x_1) \quad (4.2)$$

From the above statement, it is seen that;  $d_1(x_1)$  is the nearest distance to the  $x_1$ ,  $d_2(x_1)$  is the next nearest distance to the  $x_1$  and so on. So,  $y_1$  is the nearest neighbor of  $x_1$ ,  $y_2$  is the second nearest neighbor of  $x_1$  and so on.

4. Assign each of the test data to the class having the most examples among the  $k$  nearest neighbors.

An example of this method is given in Figure 4-1 where  $k=6$ . The distance vector is calculated and among the six nearest neighbors, three of them are members of class III. Therefore, the unknown bird is assigned to class III.

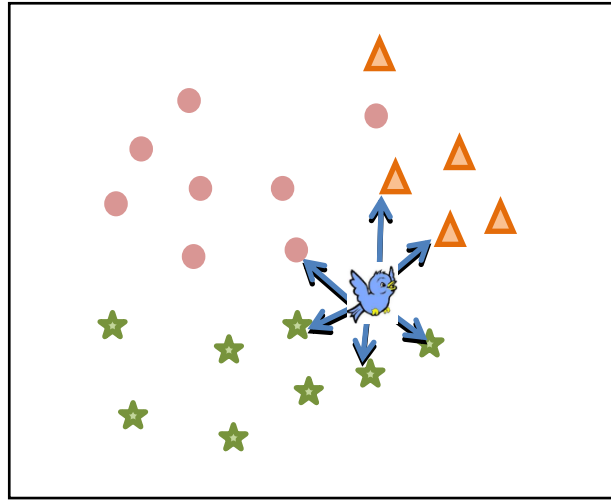


Figure 4-1:  $k$ -NN classification for an unknown bird. Dots denote class I, triangles denote class II and stars denote class III.

In this work, four different similarity measurements were used to determine the best classification results. The comparison results are given in the Chapter 5. Formulas of the similarity measurements that were used are given in Equations 4.3 - 4.6. For simplicity, equations are arranged to give the similarity between only two data,  $x$  and  $y$ . Since each data contains more than one feature (e.g. a bird call contains thirteen MFCC features); both  $x$  and  $y$  are assumed to be a vector.

- Euclidean distance: Euclidean distance gives the distance between two points in Euclidean space as:

$$d_{Euc} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.3)$$

- Mahalanobis distance: Mahalanobis distance decorrelates different features and can be calculated as:

$$d_{Mah}(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)} \quad (4.4)$$

where  $\Sigma$  is the covariance matrix of the training data.

- Cosine similarity: Cosine similarity measures the angle difference between two data vectors as:

$$sim_{cos}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\| \|y\|} \quad (4.5)$$

- Pearson correlation: Pearson correlation coefficients give a number between +1 to -1 and it measures the degree of similarity between variables x and y. Pearson's Correlation Coefficient can be expressed as:

$$sim_{corr} = \frac{\sum xy - \frac{\sum x \sum y}{N}}{\sqrt{(\sum x^2 - \frac{(\sum x)^2}{N})(\sum y^2 - \frac{(\sum y)^2}{N})}} \quad (4.6)$$

There is no written algorithm that would give the best similarity measurement for applications. Therefore the optimum similarity measurement has to be found with preliminary experiments before going further with next phases of the any application.

## 4.2 Hidden Markov Model (HMM) Classifier

Classifying unknown input data becomes easier when it is assumed that each of its elements is independent. However, this assumption cannot be used when elements of data

are dependent on each other. For instance; in English, the probability of seeing the letter h after the letter t is higher than the probability of seeing letter x in the same position. In this section, Markov models will be explained where the current state of a data is dependent on its previous state.

### 4.2.1 Discrete Markov Processes

The Markov process is a stochastic process which means that at any time, for a sequence of states  $(S_1, S_2, \dots, S_N)$ , the transition from one state to another, depends on the current state and the previous states [83]:

$$P(q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_k, \dots) \quad (4.7)$$

The state of the system at time t is symbolized as  $q_t$ . Therefore, the expression  $q_t = S_i$  states that system is in state  $S_i$  at time t. 1<sup>st</sup> order Markov models are a special case of the Markov models. In 1<sup>st</sup> order Markov models; the state at time t+1 depends on only the previous state at time t:

$$P(q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_k, \dots) = P(q_{t+1} = S_j | q_t = S_i) \quad (4.8)$$

### 4.2.2 Hidden Markov Models

In Hidden Markov Models (HMMs), the states are not observable; only the outputs are observable, which are called as observation states. Elements of HMM can be represented as:

$$\lambda = (N, M, A, B, \pi) \quad (4.9)$$

where,

- N is number of states in the model:  $S = \{S_1 S_2 S_3, \dots S_N\}$
- M is number of distinct observation symbols for each state and  $v_m$  is one of the discrete observation in the observation set
- A is the state transition probability:  $A = [a_{ij}]$ ,  $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$
- B is the observation probability:  $B = [b_j(m)]$ ,  $b_j(m) = P(O_t = v_m | q_t = S_j)$
- $\pi$  is the initial state probability vector.

N and M are implicitly defined structure parameters. Therefore,  $\lambda = (A, B, \pi)$  is the parameter set of HMM. There are three canonical problems to derive HMM:

1. Evaluating HMM: Given the parameter model of HMM,  $\lambda$ , the probability of a particular observation sequence has to be computed.
2. Decoding HMM: Given the parameter model of HMM,  $\lambda$ , and a particular observation sequence, the state sequence, which has the highest probability to generate that observation sequence, needs to be found. To find a maximum over all possible state sequences, a Viterbi algorithm is used.
3. Training HMM: Given a set of observation sequences, the most likely set of parameter models,  $\lambda$ , has to be found. A Baum-Welch algorithm is used to train the HMM, which is a special case of the Expectation-maximization procedure.

The following equations were derived from previous works in order to implement the HMM algorithm [83, 84].

### Solution of the first problem:

The forward-backward procedure is used to overcome the first problem. The forward-backward procedure consists of two phases: computation of forward variable ( $\alpha$ ) and computation of backward variable ( $\beta$ ).

Firstly, the observation sequence is divided into two parts. The first part is the forward procedure, which starts at time 1 and ends at time  $t$ . The forward variable is calculated at this part, recursively. Given the model  $\lambda$ , the forward variable is defined as:

$$\alpha_t(i) \equiv P(O_1 \dots O_t, q_t = S_i \mid \lambda) \quad (4.10)$$

The forward algorithm consists of initialization step and recursion step. The initialization step is calculated as:

$$\alpha_1(i) = \pi_i b_i(O_1) \quad (4.11)$$

The recursion step is calculated as:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad (4.12)$$

The backward procedure must now be implemented. The backward procedure starts at time  $t+1$  and continues to the end of the observation sequence. Given the model  $\lambda$  and state  $S_i$  at time  $t$ ; the backward variable is defined as:

$$\beta_t(i) \equiv P(O_{t+1} \dots O_T \mid q_t = S_i, \lambda) \quad (4.13)$$

The backward algorithm also consists of initialization step and recursion step. The initialization step is calculated as:

$$\beta_T(i) = 1 \quad (4.14)$$

And the recursion step is calculated as:

$$\beta_t(i) \equiv \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (4.15)$$

**Solution of the second problem:**

Second canonical problem aims to find the state sequence ( $Q = \{q_1 q_2 \dots q_T\}$ ) of a given model  $\lambda$ , that has the highest probability to generate the observation sequence

( $O = \{O_1 O_2 \dots O_T\}$ ). Given the observation sequence and the model,  $\gamma_t(i)$  is defined as:

$$\gamma_t(i) \equiv P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_t(j) \beta_t(j)} \quad (4.16)$$

where  $\gamma_t(i)$  is the probability of being at state  $S_i$ , at time  $t$ . To construct the state sequence, for each time  $t$ , the maximum probability ( $\gamma_t(i)$ ) is taken as:

$$q_t^* = \arg \max_i \gamma_t(i) \quad (4.17)$$

Equation 4.17 does not include the state transition conditions and a Viterbi algorithm is used to overcome this problem. Given the model and observation sequence, the highest probability of a single path at time  $t$  is expressed as:

$$\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} p(q_1 q_2 \dots q_{t-1}, q_t = S_i, O_1 \dots O_t | \lambda) \quad (4.18)$$

where the state sequence ends at state  $S_i$ . Then, starting from  $t=1$ , the optimum path is found for each  $t$ , recursively. The four main steps of the Viterbi algorithm are:

- Initialization:

$$\delta_t(i) = \pi_i b_i(O_1) \quad (4.19)$$

$$\psi_1(i) = 0 \quad (4.20)$$

where  $1 \leq i \leq N$

- Recursion:

$$\delta_t(j) = \max_i \delta_{t-1}(i) a_{ij} b_j(O_t) \quad (4.21)$$

$$\psi_t(i) = \operatorname{argmax}_i \delta_{t-1}(i) a_{ij} \quad (4.22)$$

- Termination: Termination occurs when  $t=T$  as:

$$path^* = \max_i \delta_T(i) \quad (4.23)$$

$$q_T^* = \operatorname{argmax}_i \delta_T(i) \quad (4.24)$$

- Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad (4.25)$$

Viterbi algorithms are very similar to forward-backward procedures. However, while forward-backward algorithms consider all previous states to calculate the new state at time  $t$ , Viterbi algorithm takes into account only one state, that has the maximum probability.

### **Solution of the third problem:**

Third canonical problem is related with HMM training. Training of the HMM aims to find the model parameters of the HMM. This is achieved by learning from the training



data. To find the model parameters,  $\lambda = (A, B, \pi)$ , the learning parameter ( $\varepsilon_t(i, j)$ ) is defined as:

$$\varepsilon_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (4.26)$$

The learning parameter can also be calculated as:

$$\varepsilon_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j \beta_{t+1}(j)}{\sum_k \sum_l \alpha_{kl} b_l (O_{t+1}) \beta_{t+1}(l)} \quad (4.27)$$

A Baum-Welch algorithm is used to calculate model parameters. A Baum-Welch algorithm is an Expectation Maximization (EM) process. It consists of an E-Step and an M-Step, as explained below.

- E-Step: For current model parameters,  $\varepsilon_t(i, j)$  and  $\gamma_t(i)$  are computed.
- M-Step:  $\lambda_{NEW}$  is computed by using  $\varepsilon_t(i, j)$  and  $\gamma_t(i)$  that are obtained at E-Step.
- Stop Condition: EM procedure continues until a convergence,

$$P(O | \lambda_{NEW}) > P(O | \lambda) \text{ is obtained.}$$

After the convergence, model parameters,  $\hat{\pi}_i$ ,  $\hat{a}_{ij}$  and  $\hat{b}_j(m)$  are calculated as:

$$\hat{\pi}_i = \frac{\sum_{k=1}^K \gamma_1^k(i)}{K} \quad (4.28)$$

$$\hat{a}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \varepsilon_t^k(i, j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(i)} \quad (4.29)$$

$$\hat{b}_j(m) = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(j) 1(O_t^k = v_m)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(i)} \quad (4.30)$$

The HMM implementation that was discussed previously is for discrete observation sets. In this thesis observation vectors (features of bird calls) were continuous. Therefore, Discrete Hidden Markov Models (DHMMs) were implemented for the bird call classification. Continuous observations were discretized using a k-means algorithm.

### **4.2.3 K-Means Algorithm**

K-means algorithm is a vector quantization and one of the most common unsupervised learning techniques [83]. Each data is assigned only one cluster with K-means algorithm. The algorithm divides a set of data points into non-overlapping groups of points, which are called clusters. K is a positive number and represents the number of clusters. Each cluster has different characteristics and points of one cluster carry the similar properties. Data points are grouped by calculating the distances between the data and cluster centroids. Mean squared error functions are generally used to minimize the sum of squares of distances between data and the corresponding cluster centroid.

The K-means algorithm in this work was implemented as follows:

1. Initialize K clusters (i.e.  $K=8$ ) with random points and calculate the centroids of each cluster. Centroids of each cluster are the mean of the points in the cluster.
2. Determine the distance of each observation to the centroids. Then, according to the distance value, assign each observation to the cluster that has the closest centroid.
3. Calculate the new centroids by following these steps:

- Group the observation sets based on the distances as explained in step 2. The observation set is partitioned based on the Euclidean distance from the centroids.
- After each data in the observation set is assigned to their new cluster, a new centroid of each cluster is calculated.

The algorithm stops if re-computing the centroids will result in a very small change (i.e.  $J=0.001$ ) for each centroid. This is controlled by the objective function:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (4.31)$$

where  $\|x_i^{(j)} - c_j\|^2$  is the distance measured between a data point  $x_i^{(j)}$  and the centroid of cluster  $j$ .

By using the K-Means clustering algorithm, continuous bird call observations were discretized. Then, three canonical problems of HMM were solved as explained previously.

The basic block diagram of the implemented DHMM classification process is given in Figure 4-2. At the training phase, HMM was built for each species. The number of HMM models should be equal to the number of different bird species. At the testing phase, problem 1 was solved. To recognize unknown sets of observations, the features were discretized by a k-means algorithm. Then, for each HMM model that represents the training bird species, the probability of generating the unknown observation was calculated. The model that gives the highest value when generating the input observations was the recognition result.

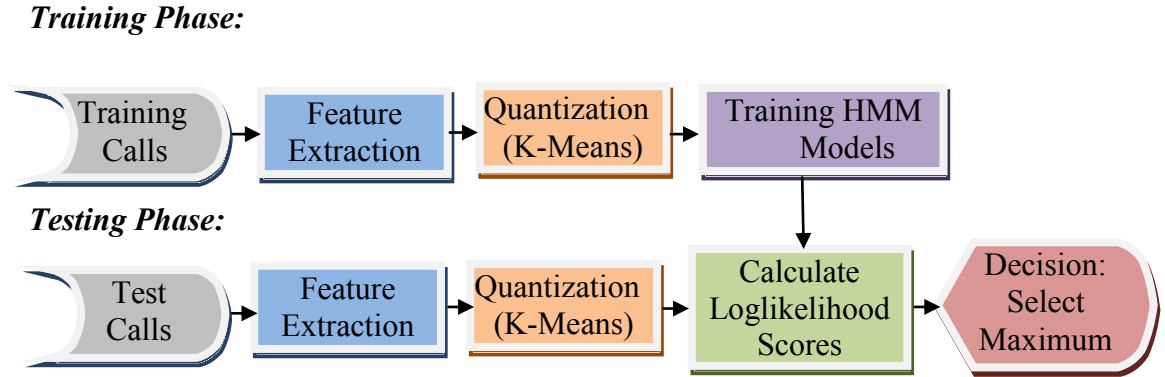


Figure 4-2: Discrete-HMM model for bird call classification

## 4.3 Multilayer Perceptron (MLP) Classifier

In this section, Multilayer Perceptron (MLP) artificial neural networks are explained. MLP networks are feedforward artificial neural networks which can be used for classification and regression [83].

### 4.3.1 Concept of Perceptron

A perceptron is a simple neuron which is the basic processing element of a neural network. Usually, practical applications of perceptrons are very limited. However, since the theoretical analysis of practically useful neural networks is usually difficult, it is more convenient to understand the theory of a simple perceptron [85].

The perceptron has multiple real-valued inputs and each input has a weight and an activation function. The output of the perceptron is obtained by linear combination of its input weights and applying a nonlinear activation function to the sum. The actual output of neuron  $j$  can be expressed as [85]:

$$y_j = f(\sum_{i=1}^n w_{ij} x_i + \theta_j) \quad (4.32)$$

where,  $w_{ij}$  is the vector of weights for the connection between neuron  $i$  and neuron  $j$ ,  $x$  is the vector of inputs,  $\theta_j$  is the bias of the neuron  $j$  and  $f()$  is the activation function. A model of a simple perceptron is shown in Figure 4-3:

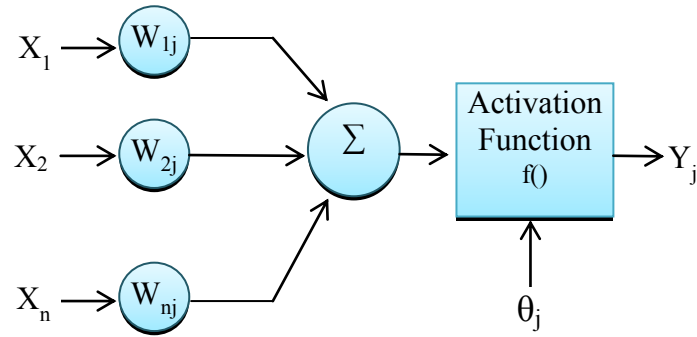


Figure 4-3: Model of a simple perceptron

Step functions were the first activation functions used in the concept of the original perceptron by Roseblatt in 1962 [86]. Step functions are expressed as:

$$f(s) = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{if } s < 0 \end{cases} \quad (4.33)$$

As seen from Equation 4.33, the output of the network is either true (1) or false (0) depending on the input. Thus, step functions are mostly used in binary classification. This is the biggest limitation of the simple perceptron. Because of its limited outputs, the simple perceptron can only classify linearly separable sets of vectors [83]. The set is linearly separable, if, and only if, output values can be separated by a line [87]. Linearly separable and inseparable sets of vectors are shown in Figure 4-3 and Figure 4-4

respectively.

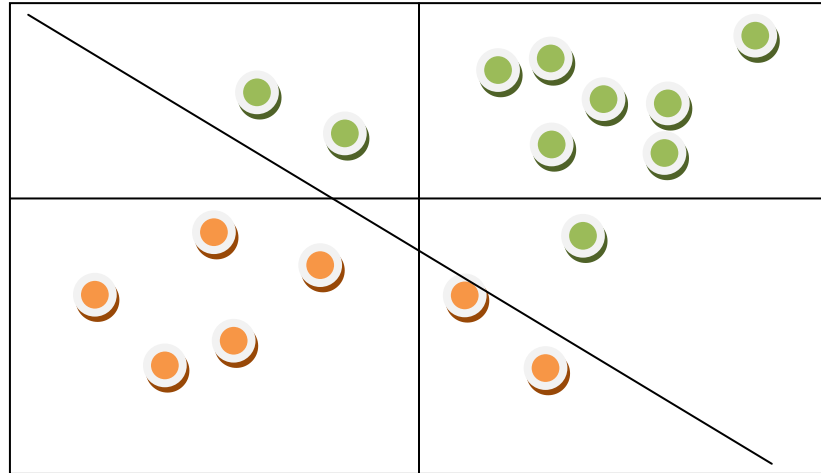


Figure 4-4: Linearly separable set

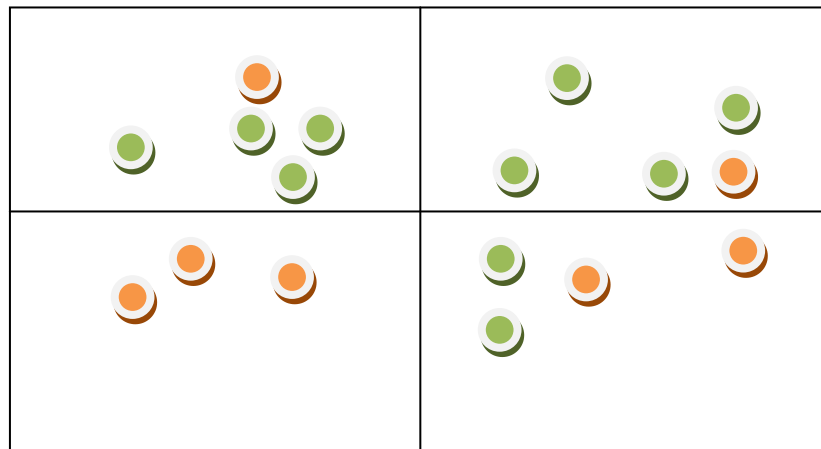


Figure 4-5: Linearly inseparable set

### 4.3.2 Multi-Layer Perceptron

The single-layer perceptron can only classify linearly separable problems. For non-separable problems it is necessary to use more layers. A multi-layer perceptron (MLP)

network has one or more hidden layers between input and output layers [83].

MLP networks are called feedforward artificial neural networks because information flows in a single direction from the input nodes to the output nodes. The network is fully connected, because every node in a layer is connected to all nodes in the next layer. If some of the links are missing, the network is partially connected. A typical architecture for a MLP multilayer neural network with one hidden layer is shown in Figure 4-6.

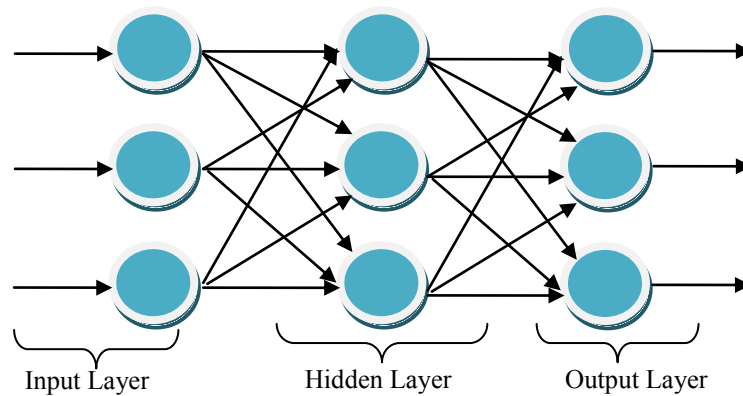


Figure 4-6: Three layer feedforward network

In Figure 4-6, the first layer is the input layer. The input layer receives the signal and passes it to the nodes in the next layer, which is a hidden layer. Each unit in the hidden layer is a neuron and applies a nonlinear activation function to its weighted sum. Then, nonlinear activation function values that are computed in hidden layers are combined at the output layer [83]. Usually, only one hidden layer is used because analysis of the network becomes more complex as the number of hidden layers increases.

Once the architecture of MLP neural network is selected, the input signals are prepared. Input signals, in this work, were the features of bird flight calls. With selected network architecture and input signals, the MLP network is trained. Training is

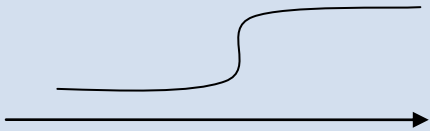
performed by a learning algorithm. The learning of the MLP network is similar to a simple perceptron. However, the output of a multilayer perceptron is a nonlinear function of the input due to the hidden layers. Also, each neuron has weights associated with its inputs, resulting in a higher number of weights to be adjusted when an error is made with a piece of training data. The most common learning technique is through the use of a supervised backpropagation algorithm to train the MLP network [83] [25]. Since network training is supervised, the network has to be provided with desired outputs for different inputs. After training is completed with various inputs, the network is tested with an unknown input set. If the network fails to classify the unknown input set, training procedures are reapplied to the network.

### 4.3.3 Backpropagation Algorithm

The role of the backpropagation training algorithm is to set the network's weights and thresholds to minimize the classification error at the testing phase.

As it was stated before, simple perceptrons use a step function as an activation function. On the other hand, MLP backpropagation networks usually use the sigmoid function [83]. The sigmoid function is a nonlinear function and defined in Table 4.1.

Table 4.1: Sigmoid function and its illustration

Sigmoid Function	
$\sigma(x) = \frac{1}{1 + e^{-x}}$	



Considering a network with one hidden layer, in which the input layer nodes are represented as  $i$ , output layer nodes are represented as  $k$  and hidden layer neurons are represented as  $j$ . Then the output value for a node  $j$  in the network is calculated as:

$$Y_j = \frac{1}{1 + e^{-x_j}} \quad (4.34)$$

where  $x_j$  is the total input of the node  $j$ .  $X_j$  is calculated as:

$$X_j = \sum_{i=1}^n x_i * w_{ij} - \theta_j \quad (4.35)$$

where  $n$  is the number of inputs to node  $j$ ,  $w_{ij}$  is the weight of the connection between each node  $i$  and node  $j$ ,  $\theta_j$  is the bias (threshold) of the neuron  $j$ ,  $\theta_j$  is the threshold value being used for node  $j$ , and  $x_i$  is the input value for input node  $i$ . Thresholds are usually set to a random value in the range [0-1].

The performance of the network is computed through the use of an error function. The error signal for the output node  $k$  is defined as:

$$E_k = D_k - Y_k \quad (4.36)$$

where  $D_k$  is the desired value and  $Y_k$  is the actual value of the node  $k$ .

Backpropagation algorithms use a gradient search technique to minimize the mean squared error value. The error gradient of the output node  $k$  is defined as:

$$\delta_k = \frac{\partial y_k}{\partial x_k} * E_k \quad (4.37)$$

where  $x_k$  is the weighted sum of the input values at node k, and y is the sigmoid function. Since y is defined as the sigmoid function of x, the derivative of the sigmoid function could be used to derive the error gradient. Sigmoid functions can easily be differentiated as:

$$\frac{d\sigma(x)}{dx} = \sigma(x) * (1 - \sigma(x)) \quad (4.38)$$

Using the above equation, the error gradient is expressed as:

$$\delta_k = y_k(1 - y_k)E_k \quad (4.39)$$

Then, the error gradient for each node j is calculated as:

$$\delta_j = y_j * (1 - y_j) * \sum_{k=1}^n w_{jk} \delta_k \quad (4.40)$$

After the error gradient value is calculated, the values of the weights are changed according to the following equations:

$$w_{ij} \leftarrow w_{ij} + (\eta)(x_i)(\delta_i) \quad (4.41)$$

$$w_{jk} \leftarrow w_{jk} + (\eta)(y_i)(\delta_k) \quad (4.42)$$

where  $w_{ij}$  and  $w_{jk}$  are weights in the network,  $\eta$  is the learning rate, and  $x_i$  is the input value to the input node i.

The procedure of backpropagation algorithm that was implemented in this work is summarized as follows:

1. Initialization: Initialize the network by setting weights to random values. Weights are usually set to small values in the range [-0.5-0.5]

2. Forward pass: Feed the input signals through the network from the input layers to the outputs. Since the weights are set randomly, the output value would be completely different than the desired value.
3. Error function: Calculate the error of each neuron in the network.
4. Backward pass: Feed calculated error values through the network. Values of weights are adjusted with the feed backward pass to reduce the error of the next time.
5. Iteration: The process is repeated in this way until the error is minimized. In other words, the algorithm stops when outputs produced for the training data are sufficiently close to the desired values.

In the testing phase, test inputs and new patterns are sent to the network for classification. Because MLP networks have randomly initialized weights, running experiments with the same parameter may yield different results from time to time.

## **4.4 Evolutionary Neural Network (ENN) Classifier**

The ENN classifier consists of feedforward neural networks and a Genetic Algorithm (GA). Feedforward Neural Networks were explained in Section 4.3.2.

### **4.4.1 Genetic Algorithm (GA)**

The learning algorithm responsible for training the neural network is a fundamental characteristic of an Artificial Neural Network (ANN). Traditional neural networks use backpropagation algorithms, such as the previously explained MLP network, for training. However, backpropagation algorithms have two main disadvantages [31]: reaching the

local minima and inefficient estimation of the topology. Therefore, a Genetic Algorithm (GA) was used to train the feedforward network in this work. GAs are very efficient in optimizing both weight and topology selections.

Genetic algorithms operate similarly to the natural evolution process when searching and optimizing a problem. In a complex, high dimensional search space, GAs search for the best solution based on the evolution. Principles of genetic algorithms were proposed first by John Holland in 1975 [88].

Instead of finding one solution, genetic algorithms produce a solution set with different solutions. By doing so, in the search space, lots of points are evaluated simultaneously. Thus, the probability of finding the best solution increases.

Genetic algorithm starts with sets of chromosomes which represent sets of solutions. Sets of chromosomes are called populations. Chromosomes are received from feedforward networks. Each chromosome is evaluated according to its fitness value. Populations of solutions are selected by using chromosomes with better fitness values. Fitness functions indicate the quality of the chromosome. Best solutions are found by applying genetic operators such as crossover and mutation. New populations are called offspring. Chromosomes that form the offspring are called parents. Steps are repeated until a termination condition is satisfied. The termination condition is satisfied if acceptable solutions have been found or computational resources have been spent. Steps of the GA are given in Table 4.2. These steps are explained in detail in the next section[89].

Table 4.2: Basic steps of the Genetic Algorithm

1. Initialization
<ul style="list-style-type: none"> <li>• Define the size of the population of chromosomes</li> </ul>
2. Fitness Function
3. Create offspring
<ul style="list-style-type: none"> <li>• Selection by using Roulette Wheel Selection</li> <li>• Apply genetic operators (crossover and mutation)</li> </ul>
4. Iteration
<ul style="list-style-type: none"> <li>• If the termination condition is satisfied: Stop and Return the best solution</li> <li>• If the termination condition is not satisfied: Go to step 2</li> </ul>

#### 4.4.2 ENN Algorithm

ENN algorithms consist of training and testing phases. In the training phase, feedforward networks are trained by genetic algorithms. A block diagram of an ENN algorithm is given in Figure 4-7.

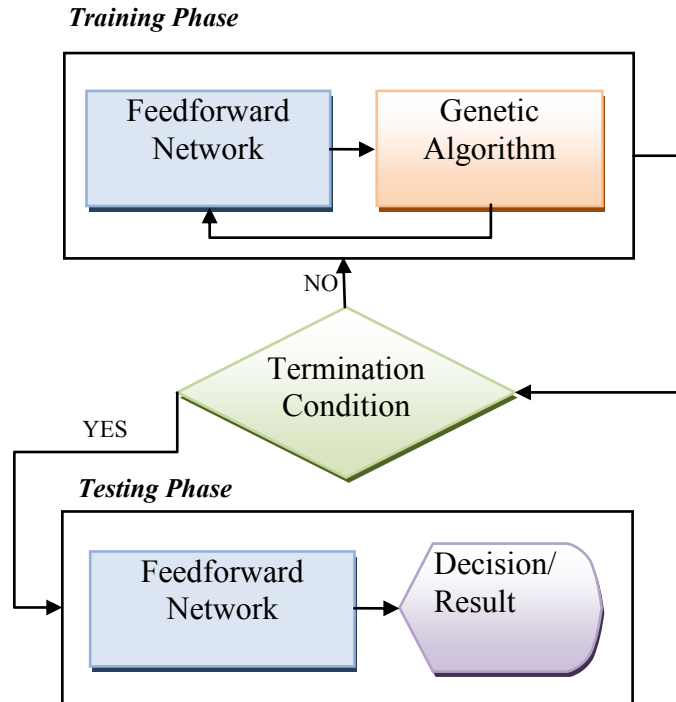


Figure 4-7: Block diagram of Evolutionary Neural Network

Training of the ENN algorithm was performed using following steps:

Initialization: The following were defined at the initialization step:

- The number of chromosomes
- The number of generations
- Crossover probability
- Mutation probability
- Number of training and test data
- Initial random population of chromosomes
- The initial weights

Initial weights were set to small values between -1 to 1. Then, individual chromosomes were applied to the feedforward neural network.

Fitness Function: After initialization, the fitness value of each individual chromosome

was calculated. Fitness values of chromosomes are usually calculated by fitness functions, using the mean square error of output neurons in the training set [90]. Fitness functions can be defined as:

$$Fitness = \frac{1}{E_i^2} \quad (4.43)$$

Where  $E_i$  is the mean square error function of the network [91]. Error functions of the network are calculated as:

$$E_i = \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^n (y_j^k - y_j)^2 \quad (4.44)$$

Where

- $m$  is number of samples of practice cluster
- $n$  is the number of neurons in the input layer
- $y_j^k$  is the desired output of neuron  $j$  in output layer
- $y_j$  is the actual output of the output layer of neuron  $j$

Each of the chromosomes was then assigned a fitness value. After the fitness evaluation, chromosomes with poor fitness are eliminated from the population. The actual output of neuron  $j$  can be expressed as:

$$y_j = f(\sum_{i=1}^n x_i w_{ij} + b_j) \quad (4.45)$$

Where,  $x_i$  is the input signal,  $w_{ij}$  is the weight of the connection between neuron  $i$  and neuron  $j$ ,  $b_j$  is the bias of neuron  $j$ , and  $f(.)$  is the activation function. Sigmoid

functions were used as activation functions. Sigmoid functions were defined in Table 4.1.

**Roulette Wheel Selection:** In every reproduction step, the offspring is created from the parent chromosomes. Different types of reproduction techniques result in a different number of new populations. The most common selection technique is the Roulette Wheel Selection. In this work, also, The Roulette Wheel Selection scheme was used for selecting the best individual chromosomes from the initial population. The size of the initial population depends on the application [89]. Parent chromosomes are chosen based on the fitness values of each chromosome in the search space. Chromosomes with better fitness values are selected to create the offspring. If a roulette wheel which is divided into portions, each chromosome is represented by one portion, and each portion has a different volume proportional to the fitness value of the corresponding chromosome. The wheel is then spun, randomly selecting the parent chromosome. Therefore, selection is highly proportional to the fitness values. The probability of being selected, for each individual, is calculated as:

$$P_i = \frac{F_i}{\sum_i F} \quad (4.46)$$

where  $P_i$  is the probability and  $F_i$  is the fitness value of the corresponding chromosome.

**Genetic Operators:** Genetic operators were applied to the selected chromosomes. Crossover and mutation are two fundamental operators of the genetic algorithm [89]. Genetic operators can create better characteristic offspring from parents.

A crossover operator aims to produce an offspring by combining the genes of two parent chromosomes. Recombination processes are performed according to the crossover



probability and crossover points. Operation starts by using two parent chromosomes which are selected using Roulette Wheel Selection. An example of a single-point crossover is shown in Figure 4-8.

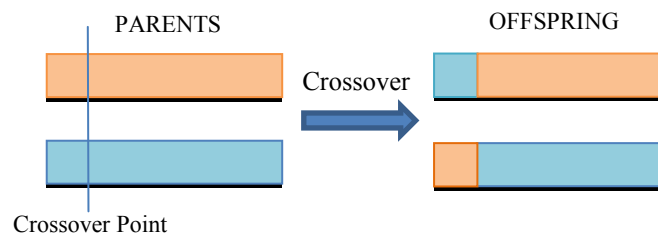


Figure 4-8: Single-point crossover operation

Mutation operator aims to create a new chromosome by replacing its genes randomly. The constant process of creating a new population, using the same chromosomes of the current population, can result in the same chromosomes after some time. This problem is known as local minimum. Mutation operators overcome this problem by increasing the variety in the population. The genes of a chromosome are replaced according to the mutation probability and mutation point. An example of a mutation operator is shown in Figure 4-9.

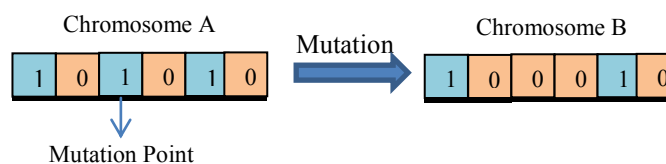


Figure 4-9: Mutation Operation

Iteration: After genetic operators were performed, the termination condition was

checked. Termination condition is a stopping criterion. If the termination condition is satisfied, the best solution is returned and the training phase of the ENN algorithm ends. If the termination condition is not satisfied, the genetic operators are reapplied to the chromosomes to find the best solution. According to the termination condition, the size of the offspring must be equal to the size of the initial population. The network should be able to classify the unknown test objects after training is completed.

After the training process, the network was tested with new feature sets. The parameters of the ENN algorithm used in this work are given in Chapter 5.

In this chapter, four different classifiers were built to recognize bird species from their features. Using these classifiers with four feature extraction methods resulted in sixteen recognition results for each species which will be given in the next chapter.

# **Chapter 5**

## **Data Collection and Simulation**

### **5.1 Data Collection**

Flight calls were recorded using Wildlife Acoustic's Song Meter SM2 night flight call package [60], as shown in Figure 5-1. The SM2 night flight call package consists of a SM2 recorder platform and a SMX-NFC microphone. The SMX-NFC microphone is waterproofed and specially designed to record distant night flight calls. The flat horizontal surface, on which the microphone capsule mounted, creates a pressure zone on the surface. Specifications of the SMX-NFC microphone are given in Table 5.1.

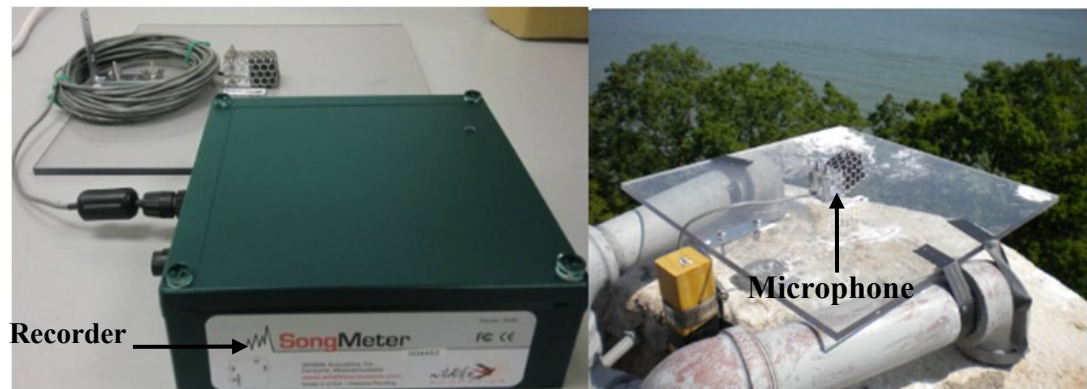


Figure 5-1: Wildlife Acoustics SM2 recorder

Table 5.1: Microphone specifications

SMX-NFC Microphone	
Frequency Response:	11kHz
Signal Gain:	3-6 dB
Beam Angle:	125 degrees
Type:	PZM

Data was collected at three different locations, during the spring migration, between April and June of 2011 as shown in Figure 5-2.

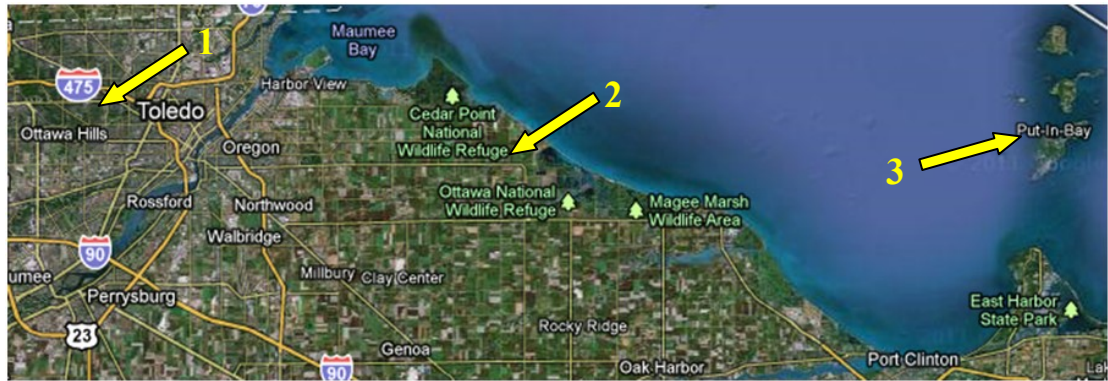
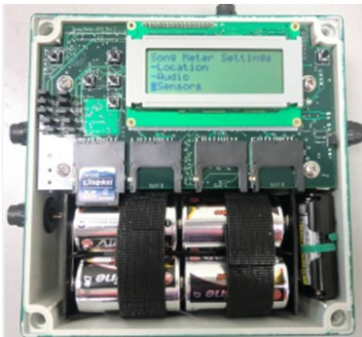


Figure 5-2: Google satellite view of the project area in Ohio, USA  
 1 = University of Toledo (Toledo, OH),  
 2 = Ottawa National Wildlife Refuge (Oak Harbor, OH),  
 3 = Ohio State University's Stone Lab (Put-in-Bay, OH)

The configuration of each SM2 recorder was set using the Song Meter Configuration Utility 2.2.5 software from Wildlife Acoustics [61]. The settings, for spring migration 2011 were selected as shown in Figure 5-3.



Song Meter Configuration Utility 2.2.6  
Settings:

- Model: SM2
- Schedule: Daily
- Start: 21:50
- Duration: 7 hours
- Sample Rate: 22050 Hz
- Channels: Mono-R

Figure 5-3: Configuration of the SM2 recorder

Recordings started, every day, an hour after sunset and stopped two hours before sunrise resulting in approximately 1GB of data being collected from each recorder per day. 459 calls (collected during spring migration) were used for testing. Training calls were obtained from Old Bird Inc. [57]. Nocturnal flight calls that were analyzed in this thesis come from the following species: American Redstart (AMRE), Common Nighthawk (CONI), Savannah Sparrow (SAVS), Swainson's Thrush (SWTH) and Tennessee Warbler (TEWA). Species, number of training calls and test calls used are given in Table 5.2.

Table 5.2: Training and test calls that are used in this thesis

<b>Species</b>	<b>Class</b>	<b># of Training Calls</b>	<b># of Test Calls</b>
AMRE	Warbler	20	52
CONI	Goatsuck	17	124
SAVS	Sparrow	20	50
SWTH	Thrush	34	166
TEWA	Warbler	37	67

## 5.2 Simulation

A Graphical User Interface (GUI) was created using MATLAB giving the user the ability to choose the database between the created databases and use necessary feature extraction methods and classifiers. A system flow diagram for the GUI is shown in Figure 5-4.

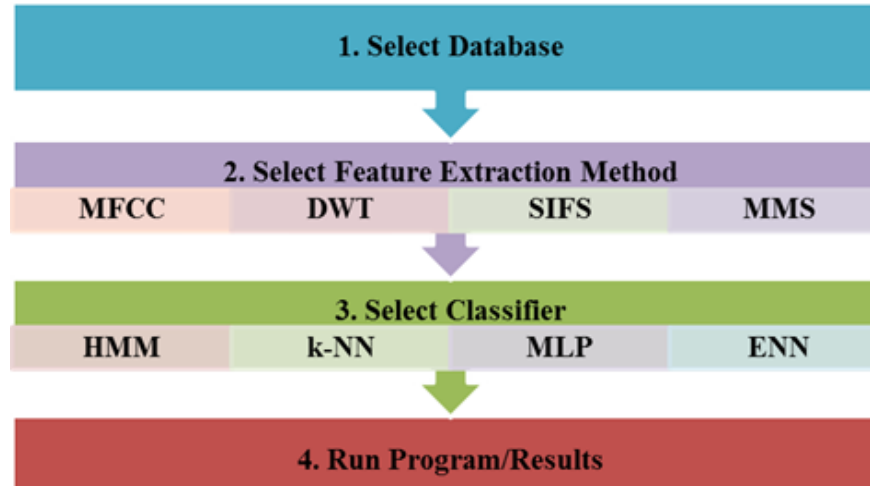


Figure 5-4: Flow diagram of the overall system and MATLAB GUI

The GUI provides results in different forms with options allowing the user to see the signal in time domain, the signal's spectrogram, the real class of the signal, and the class in which the signal was classified. Also, it allows the user to listen to the signal and start a new experiment. An example snapshot of GUI is shown in Figure 5-5.

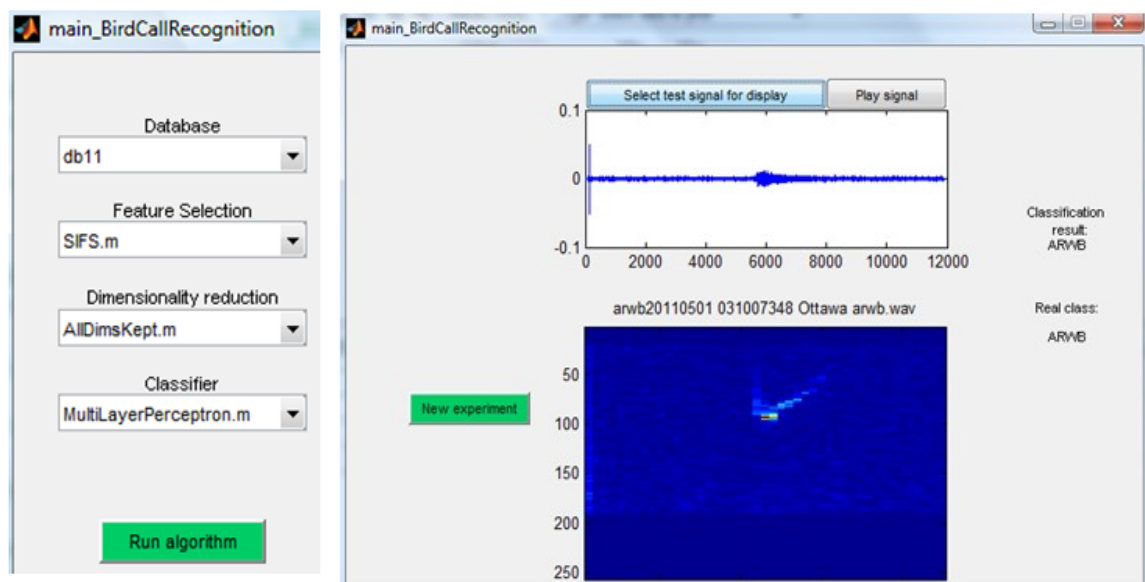


Figure 5-5: Snapshot of MATLAB GUI

All list boxes are updated automatically (after restarting the GUI) if new databases or methods are added to the respective folders.

### **5.3 Experiment Setups and Results**

Experiments were performed to test the performance of MFCC, DWT, SIFS and MMS feature vectors. MFCC features were extracted with a 20 msec Hamming window with 50 % overlapping. Twenty four triangular filters were used and the first twelve coefficients were taken from each frame in order to generate the feature vector. Wavelet coefficients were extracted by using 6-level decomposition and a db10 wavelet function. For each call, fifty four features were extracted. Feature vectors carried the information of the range, maximum, minimum, mode, median, mean, mean absolute deviation, median absolute deviation and standard deviation of the detailed coefficients. SIFS and MMS features were extracted from each call as explained in Chapter 3. Extracted feature vectors were then sent to the classifiers: HMM, k-NN, MLP and ENN respectively.

The MATLAB Bioinformatics Toolbox was used to implement k-NN classification. The recognition performance of the system was tested with a different number of neighbors and different measure metrics. The best results were obtained when k was set to 11 and the distance function was set to correlation. Performance of each similarity measurement was obtained for each feature extraction scheme. Results are shown in Table 5.3.



Table 5.3: Comparison of the similarity measurements of k-NN algorithm

	<b>kNN-Euclidean</b>	<b>kNN-Mahalanobis</b>	<b>kNN-Cosine</b>	<b>kNN-Correlation</b>
<b>MFCC:</b>	76%	71%	78%	80%
<b>DWT:</b>	70%	65%	76%	82%
<b>SIFS:</b>	73%	62%	79%	85%
<b>MMS:</b>	84%	73%	83%	89%

MLP networks were constructed with two hidden layers. Fifty neurons were in the first hidden layer and twenty neurons in the second hidden layer. Data was randomly divided into train, validation and test sets. Training was stopped after the network's error on the validation set was not reduced. Performance was measured as the mean-squared error of the network. The performance graph of a MLP is shown in Figure 5-6. The best validation performance is achieved at epoch (iteration) 10 as 0.10406. Since the validation and test curves are very similar, the network does not have an over-fitting problem.

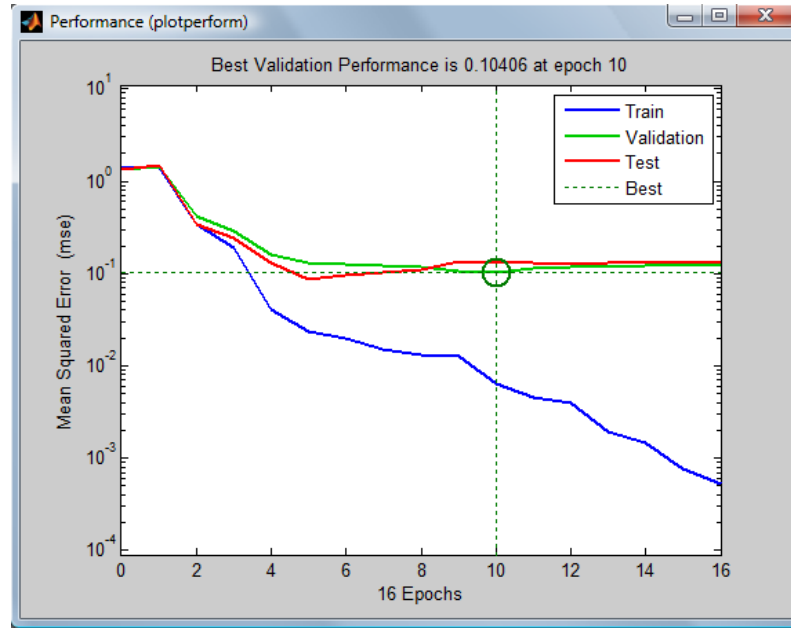


Figure 5-6: Performance graph of MLP when MMS feature extraction scheme was used

HMM algorithm was implemented using equations in Chapter 4. Five classifiers were built since there were five different species in the training database. For each training process, iteration had been continued until the difference between the newlikelihood and oldlikelihood values was obtained as 0.001. Configurations of HMM classifiers for each of the feature extraction methods were set as shown in Table 5.4.

Table 5.4: HMM configurations for flight call classification

	MFCC	DWT	SIFS	MMS
<b>Initial elements</b>	random	random	random	random
<b>Number of Iterations</b>	126	765	282	400
<b>Number of Clusters</b>	8	8	8	8
<b># of States</b>	2	10	5	8

Configurations of ENN classifiers for each of the feature extraction methods were set as shown in Table 5.5. For each case only one hidden layer was used and the activation function was selected as sigmoid.

Table 5.5: ENN configurations for flight call classification

MFCC			DWT		
# of Chromosomes	Neurons in Hidden Layer	# of Generations	# of Chromosomes	Neurons in Hidden Layer	# of Generations
200	100	1000	200	100	1000
400	200	2000	400	200	2000
600	300	3000	600	300	3000
800	400	4000	800	400	4000
<b>1000</b>	<b>500</b>	<b>5000</b>			
SIFS			MMS		
# of Chromosomes	Neurons in Hidden Layer	# of Generations	# of Chromosomes	Neurons in Hidden Layer	# of Generations
200	100	1000	200	100	1000
400	200	2000	400	200	2000
600	300	3000	600	300	3000
<b>800</b>	<b>400</b>	<b>4000</b>	<b>800</b>	<b>400</b>	<b>4000</b>
1000	500	5000	1000	500	5000

	MFCC	DWT	SIFS	MMS
# of Inputs	12	54	16	28
# of Outputs	5	5	5	5
Crossover Probability	0.7	0.7	0.7	0.7
Mutation Probability	Random	Random	Random	Random

As it is seen from Table 5.5, for MFCC and DWT; the best performance was obtained with 1000 chromosomes with 500 neurons within hidden layer after 5000 generations. For SIFS and MMS; the best performance was obtained with 800 chromosomes with 400 neurons within hidden layer after 5000 generations. Best recognition results were obtained when MMS features were used. The mean square error graph of ENN with MMS features is shown in Figure 5-7. The error for the random weights is large at the beginning of the training process and decreases as the training continues.

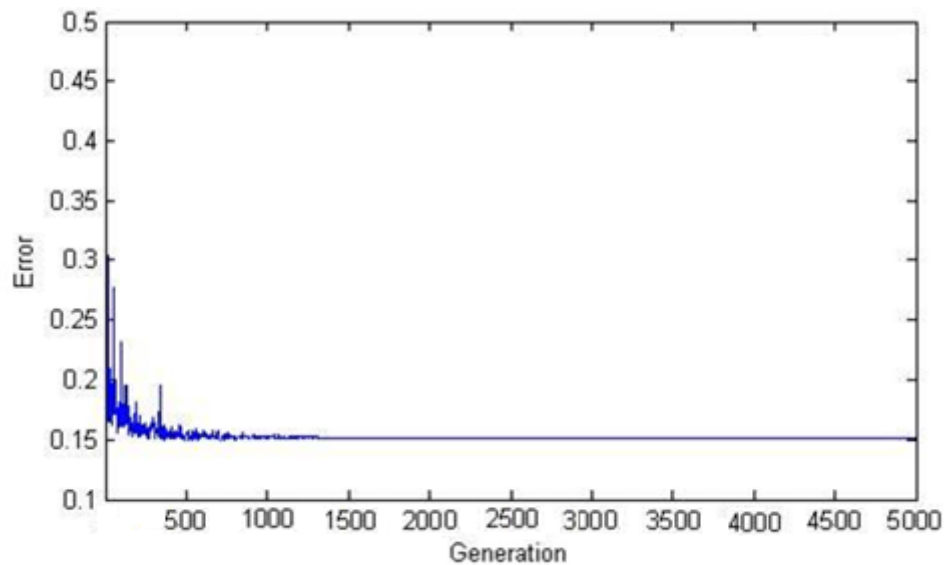


Figure 5-7: Training performance of the ENN with MMS features

Correct classification percentages for each species and the total correct classification

percentages of 459 test calls are given in Table 5.6.

Table 5.6: Percentages of classification accuracy for each feature extraction scheme

MFCC					DWT				
	HMM	kNN	MLP	ENN		HMM	kNN	MLP	ENN
	(%)	(%)	(%)	(%)		(%)	(%)	(%)	(%)
ARWB	58	65	63	67	ARWB	54	66	62	65
CONI	69	93	74	92	CONI	72	86	73	88
SASP	54	68	60	70	SASP	60	82	61	83
SWTH	70	80	82	84	SWTH	85	87	92	91
TEWB	59	75	68	77	TEWB	47	73	80	82
TOTAL	65	80	73	82	TOTAL	69	82	78	85
SIFS					MMS				
	HMM	kNN	MLP	ENN		HMM	kNN	MLP	ENN
	(%)	(%)	(%)	(%)		(%)	(%)	(%)	(%)
ARWB	65	65	57	67	ARWB	67	73	79	80
CONI	76	89	98	94	CONI	79	95	95	96
SASP	62	84	84	85	SASP	70	80	82	83
SWTH	77	90	96	87	SWTH	80	90	97	84
TEWB	66	79	85	87	TEWB	80	84	89	90
TOTAL	72	85	90	86	TOTAL	78	89	92	91

Figure 5-8 shows performance of feature extraction methods. It can be seen that MMS features outperform MFCC, DWT and SIFT features when HMM, k-NN, MLP and ENN are used as classifiers.

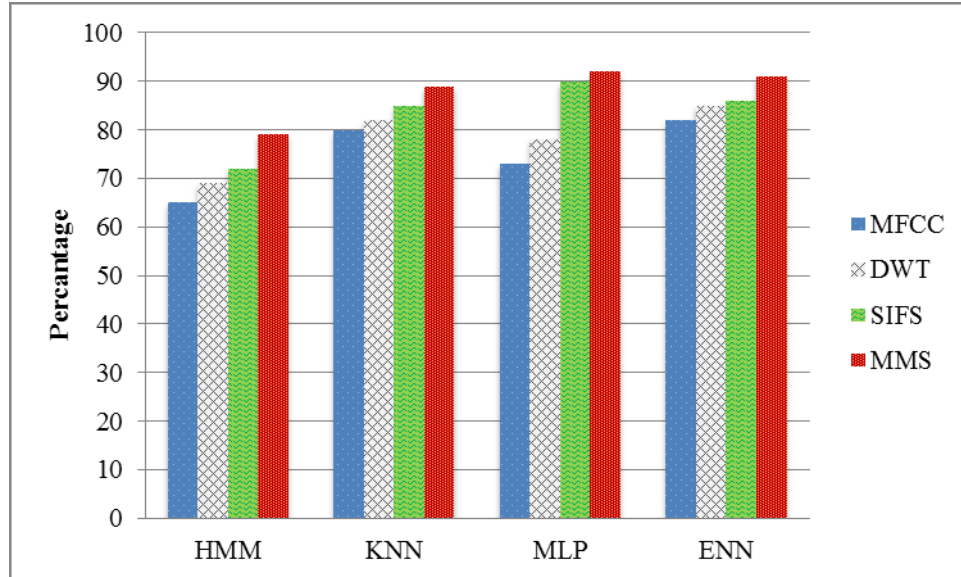


Figure 5-8 Performance of Feature Extraction methods

### **Song Scope Implementation**

The recognizer for each species is created by setting the parameters of Song Scope as shown in Figure 5-9.

	AMRE	CONI	SAVS	SWTH	TEWA
TRAINING CALLS					
Number of recordings	20	17	20	34	37
Number of annotations used	11	19	14	13	14
PARAMETERS					
Sample Rate (Hz)	22050	22050	22050	22050	22050
FFT Size	256	256	256	256	256
FFT Overlap	0,5	0,5	0,5	0,5	0,5
Frequency Min (bins)	45	14	70	14	58
Frequency Range (bins)	57	45	60	45	70
Background Filter	1s	1s	1s	1s	1s
Max Syllable (msec)	30	60	20	60	30
Max Syllable Gap (msec)	20	20	20	20	20
Max Song (msec)	90	100	70	100	90
Max complexity	32	40	32	30	32
Minimum resolution	6	20	6	4	6
Minumum quality	10	10	10	10	10
Minimum score	50	50	50	50	50
RECOGNIZER INFORMATION					
Cross training	74.42 ± 10.92	79.44 ± 3.55%	82.83 ± 5.70%	75.56 ± 4.61%	84.52 ± 4.01%
Total training	70.31 ± 10.24	80.20 ± 3.80%	81.57 ± 5.03%	80.96 ± 5.18%	82.05 ± 4.47%
Model states	17	36	25	27	26
State usage	4 ± 4	7 ± 2	8 ± 1	5 ± 3	6 ± 1
Feature vector	4	20	6	4	6
Mean symbols	6 ± 5	20 ± 7	8 ± 2	8 ± 12	7 ± 1
Syllable types	2	10	3	4	6
Mean durations	0.21 ± 0.06s	0.21 ± 0.03s	0.08 ± 0.01s	0.21 ± 0.08s	0.08 ± 0.08s

Figure 5-9: Song Scope parameter and recognizer information for each species (training)

Results from Song Scope implementation are given in Table 5.7. Song Scope gives a total of 70% correct identification results.

Table 5.7: Results of Song Scope Implementation

Species	Test Calls	Correctly Identified	Percentage (%)
ARWB	52	46	88
CONI	124	110	89
SASP	50	30	60
SWTH	127	102	80
TEWB	67	32	48
Total	459	320	70

Figure 5-10 shows the overall accuracy of four classifiers when MMS features were used alongside results of the commercial software, Song Scope. It is shown that MLP has the highest correct classification rate of 92 %.

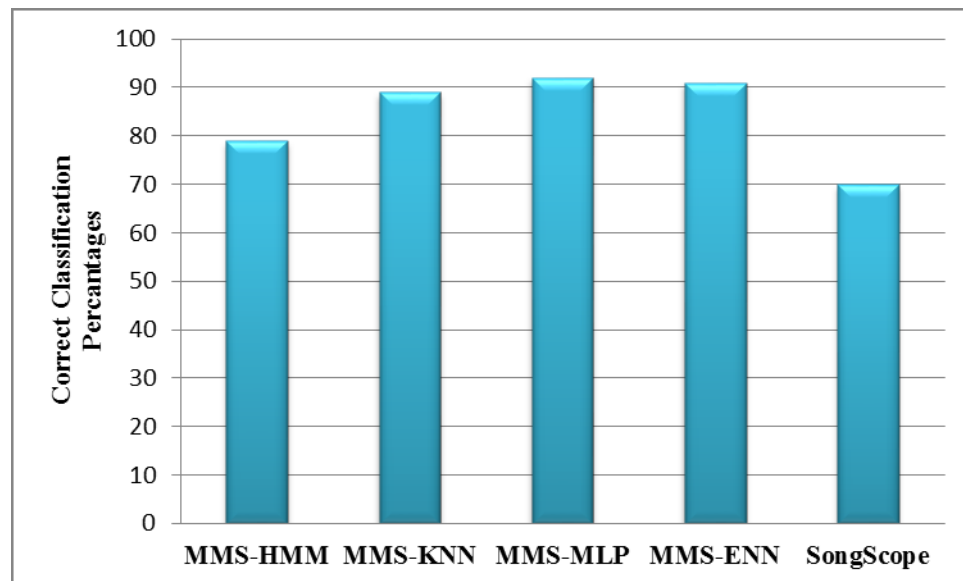


Figure 5-10: Performance comparison of the classifiers when MMS feature extraction is used



## **5.4 Quantification of Class Level Migration In NW Ohio**

### **Spring 2011**

Flight calls were recorded using Wildlife Acoustic's Song Meter SM2 night flight call package during 2011 Spring bird migration period between April and June of 2011. Data was collected at three different locations: University of Toledo (Toledo, OH), Ottawa National Wildlife Refuge (Oak Harbor, OH), and Ohio State University's Stone Lab (Put-in-Bay, OH)

The Minimum Individual Passing (MIP) [55] method was applied to obtain the total number of birds which were then grouped into three classes, rather than being classified at the species level. According to the MIP method, for thrushes, calls which were more than two minutes apart were assumed to have come from different individuals. For thrushes and warblers, however, calls which were more than one minute apart were assumed to have come from different individuals.

The thrush-class calls were in the 2-4 kHz range and their durations were typically less than 300 ms. In this thesis, thrush calls were mainly species of Gray-cheeked Thrush, Hermit Thrush, Swainson's Thrush, Wood Thrush and Veery. The sparrow and warbler class calls were in the 5-11 kHz range and their call durations were observed to be less than 150 ms. The observed sparrow and warbler calls were mainly from the following species: Fox Sparrow, Grasshopper Sparrow, Savannah Sparrow, White-crowned Sparrow, White-throated Sparrow, Black and White Warbler, Palm Warbler, Tennessee Warbler, Yellow Warbler and Yellow-rumped Warbler. The number of the thrushes, warblers and sparrows, calculated using the MIP method from three locations

between April 20 – May 29 are given in Figures 5-10 thru 5-13, respectively.

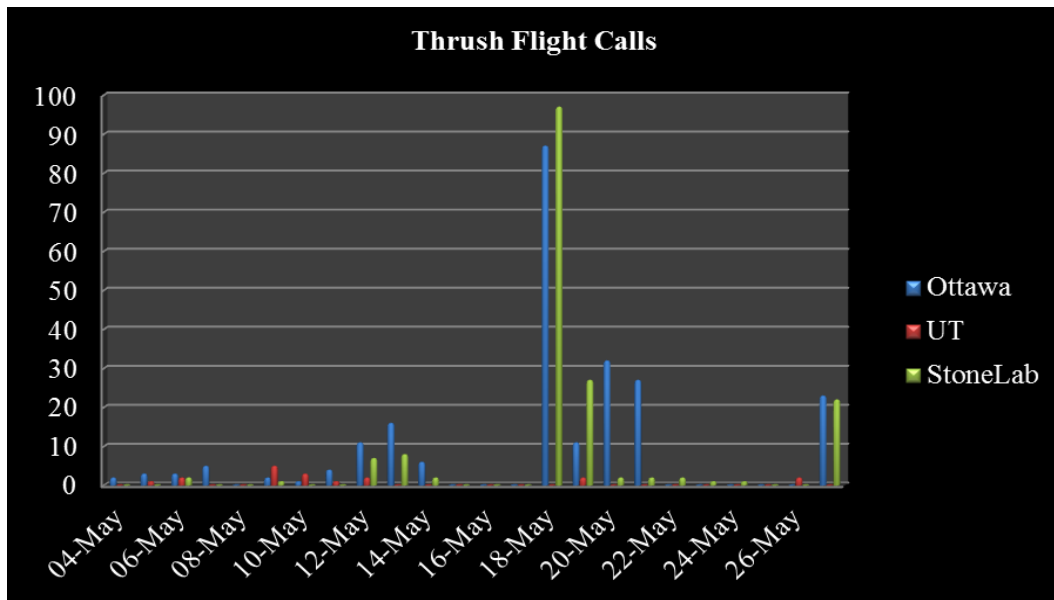


Figure 5-11: Average number of total thrushes detected at three locations during the spring migration of 2011

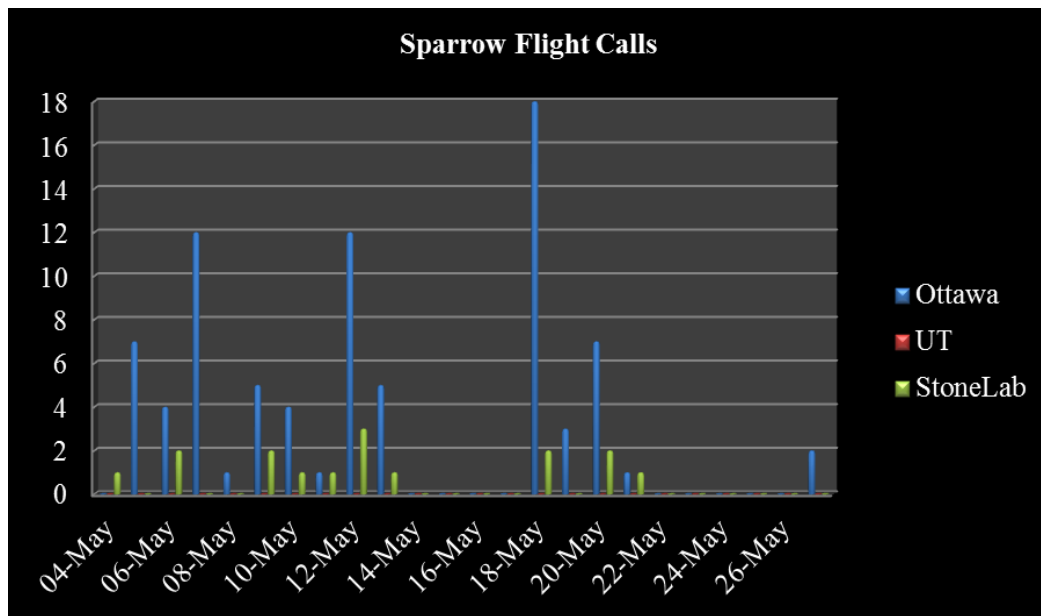


Figure 5-12: Average number of total sparrows detected at three locations during the

### spring migration of 2011

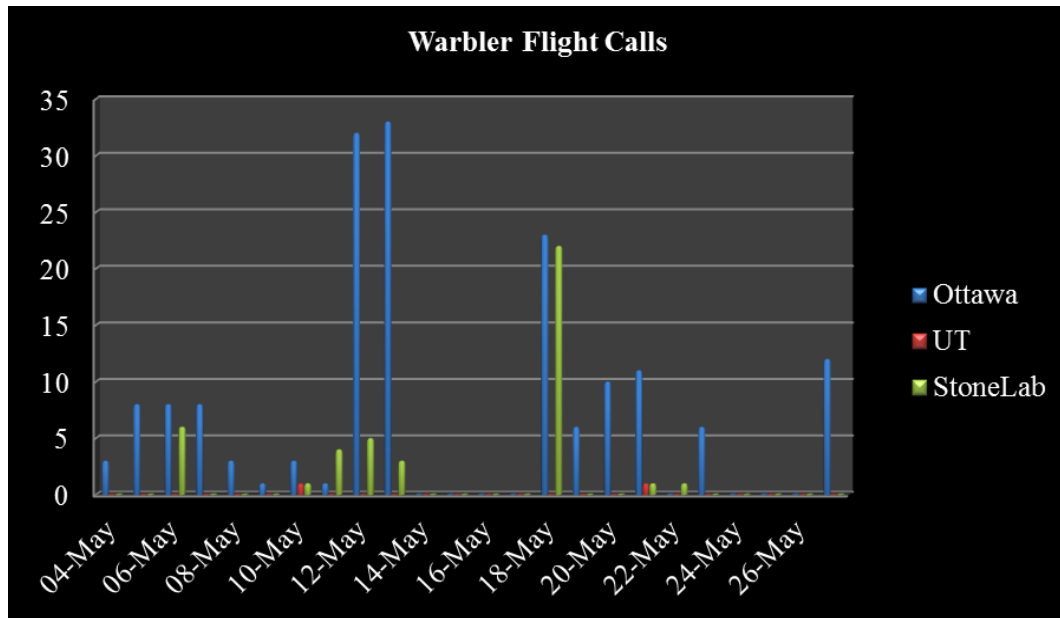


Figure 5-13: Average number of total warblers detected at three locations during the spring migration of 2011

More thrush calls were detected than both warbler and sparrow calls. This was expected as thrushes are a low-flying species which results in their calls being louder and more easily detectable. The distribution, both between and within bird classes, varied greatly from location to location. Very few birds were detected at the University of Toledo location because this location does not lie in the migration path of these birds. Furthermore, at the Putt-in-Bay location, very few sparrows were detected since they are a physically smaller class of birds which is not prone to flying over large bodies of water.

### Autumn 2011

Flight calls were recorded using Wildlife Acoustic's Song Meter SM2 night flight call

package during 2011 fall bird migration period between August and October of 2011.

Data was collected at three different locations: Toledo (Toledo, OH), Ottawa National Wildlife Refuge (Oak Harbor, OH), and Ohio State University's Stone Lab (Put-in-Bay, OH). The number of the thrushes, warblers and sparrows, calculated using the MIP method from three locations between August 31 – September 28 are given in Figures 5-14 thru 5-19 respectively.

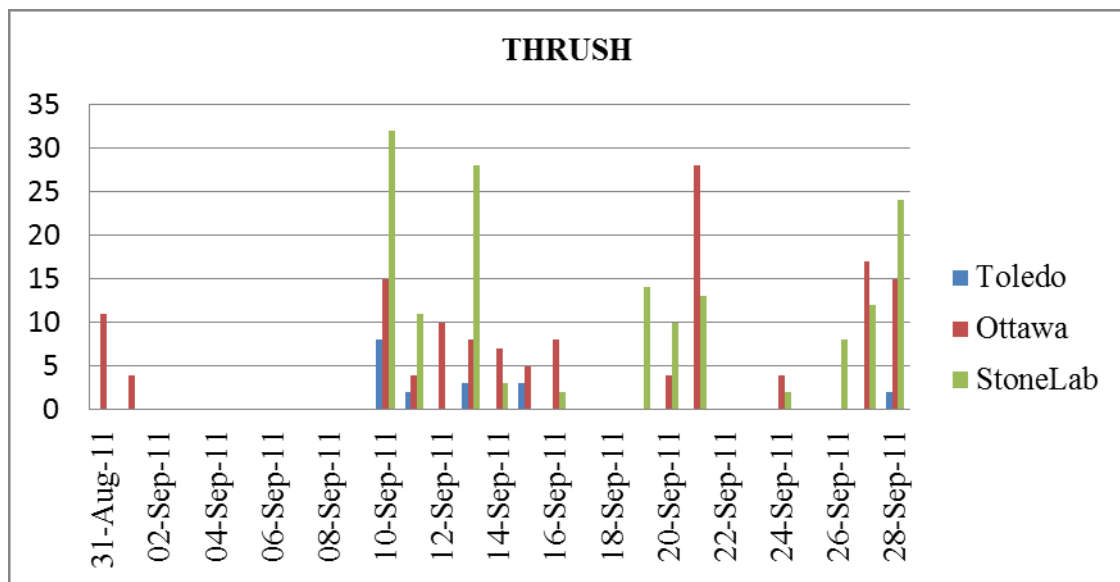


Figure 5-14: Average number of total thrushes detected at three locations during the fall migration of 2011

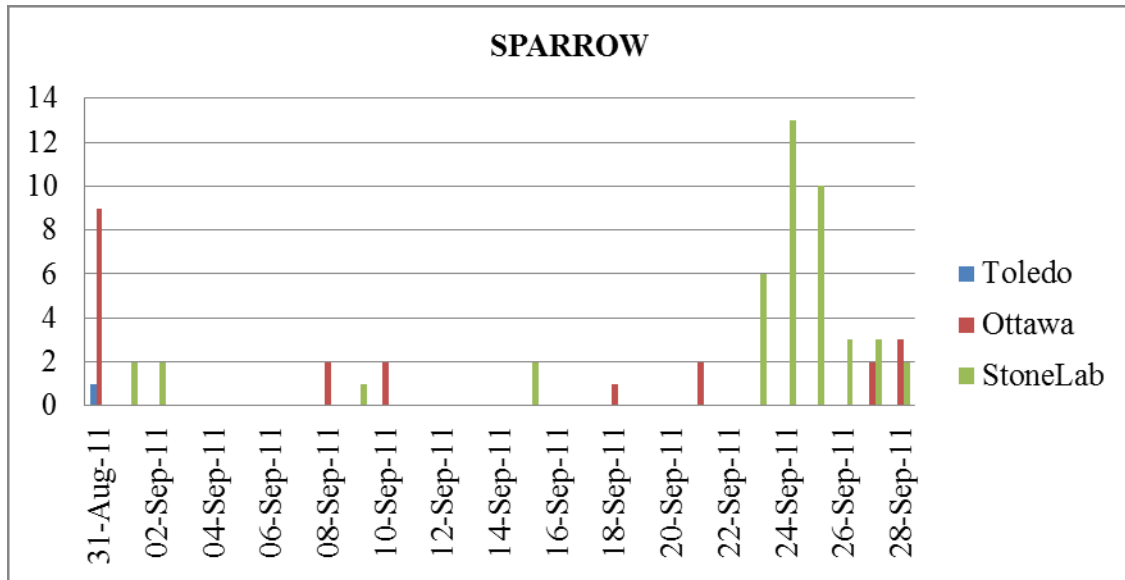


Figure 5-15: Average number of total sparrows detected at three locations during the fall migration of 2011

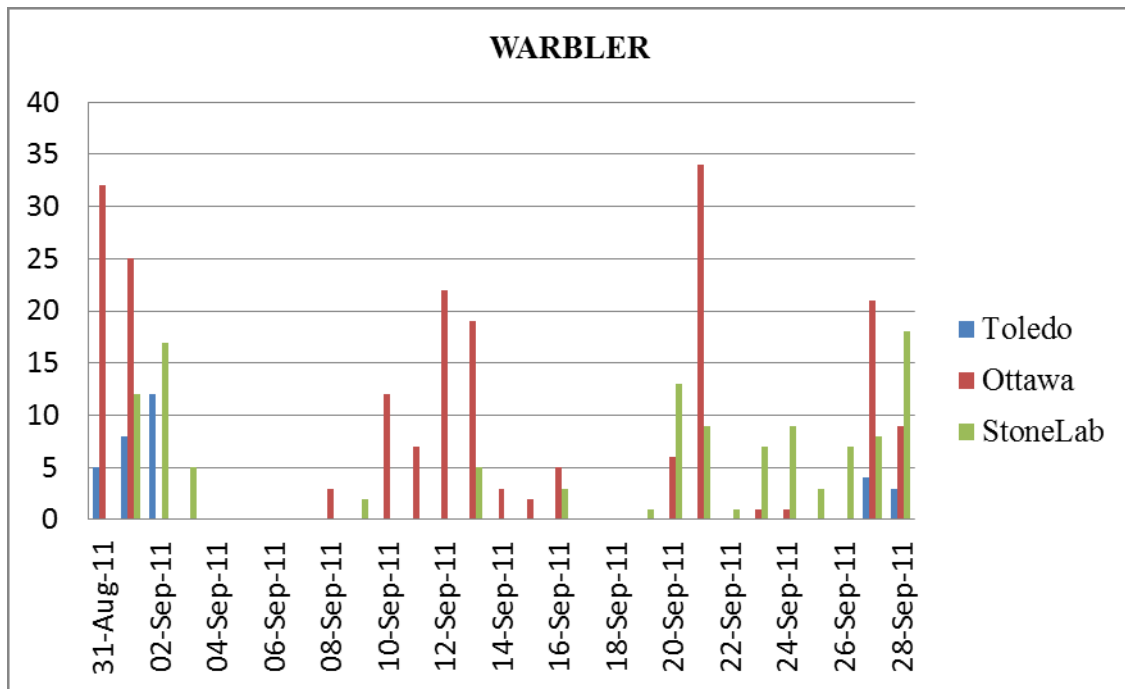


Figure 5-16: Average number of total warblers detected at three locations during the fall migration of 2011

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

Identifying and quantifying migratory bird calls, as well as comparing their call counts, from different locations provides wildlife biologists with valuable information when assessing the behavior of birds in any area, and especially in the vicinity of wind turbines. This thesis investigated the capability of different recognition systems for identification of bird species based on audio recordings of nocturnal flight calls.

For five species, flight call databases have been developed by using a commercially available CD [57]. Data was collected at three different locations during the spring migration 2011. In this thesis, four feature extraction schemes and four classifiers were built by using MATLAB software with the aim of identifying these five species from the data collected. Also, commercially available bird sound recognition software, Song Scope, was used for comparison purposes, yielding seventeen experiments for the recognition of birds based on their flight call.

Syllables were represented with different numbers of acoustic features with different

feature extraction schemes. Twelve MFCC features, fifty-four DWT features, sixteen Spectrogram-based Image Frequency Statistics (SIFS) and twenty eight Mixed MFCC & SIFS Features (MMS) were extracted for each flight call. These features were fed into the following classifiers: k-NN, HMM, DWT and ENN. All classification results from the new feature extraction schemes (SIFS and MMS) provided better results when compared to traditional MFCC and DWT techniques. Classification results from the MMS feature extraction scheme with MLP classifier provided the best result which was a 92 % recognition rate. The ENN classifier is relatively a new classifier in the application of bird species identification and results were promising. Better recognition rates with ENN classifiers may be obtained with large number of chromosomes and generations. However, in that case, computational time would also be increased dramatically. Still, recognition results were also better than the commercially available Song Scope software which had 70 % total recognition rate.

This research will help biologists in developing mitigation techniques and will also help in the development of public policy in regard to the impact of wind farms on bird populations.

## **6.2 Future Work**

Recording in different places results in inconsistent data as environmental conditions and noise distributions are not the same. Furthermore, recordings, taken at the same location, may differ if the environmental conditions are not the same. Also, the same species in different regions do not always have the same flight call. A detection and recognition system needs to be invariant for such differences. Therefore, training call

databases should contain a high number of calls, for the species of interest, from different geographical regions. The flight call database used in this thesis was obtained from a commercially available CD and not collected from three locations where test calls were recorded. For this reason, the flight call database could be increased with the new flight calls that were recognized in this thesis. This would make the recognition system more robust and powerful for the next recording's analysis in this region.



# References

1. Farnsworth, A., M.E. Powers, R.S. Hames, and K.V. Rosenberg, Final Report for Legacy Program: Migratory Bird Monitoring Using Automated Acoustic and Internet Technologies. 2007, Department of Defense Legacy Resource Management Program Project
2. Clemmons, J. and R. Buchholz, Behavioral Approaches to Conservation in the Wild. *The Journal of Wildlife Management*, 1997. 104: p. 382.
3. Brandes, T.S., Automated sound recording and analysis techniques for bird surveys and conservation. 2008. p. S163–S173.
4. Marquis, R.J. and C.J. Whelan, Insectivorous Birds Increase Growth of White Oak through Consumption of Leaf-Chewing Insects. *Ecology*, 1994. 75(7): p. 2007-2014.
5. Howe, H.F. and J. Smallwood, Ecology of Seed Dispersal. *Annual Review of Ecology and Systematics*, 1982. 13: p. 201-228.
6. Stiles, F.G., Ecological and Evolutionary Implications of Bird Pollination. *American Zoologist*, 1978. 18: p. 715-727.
7. Sodhi, N.S., COMPETITION IN THE AIR : BIRDS VERSUS AIRCRAFT. *The Auk*, 2002. 119: p. 587-595.
8. Osborn, R.G., K.F. Higgins, R.E. Usgaard, C.D. Dieter, and R.D. Neiger, Bird Mortality Associated with Wind Turbines at the Buffalo Ridge Wind Resource Area,

- Minnesota. *The American Midland Naturalist*, 2000. 143(1): p. 41-52.
9. Nicholson, C.P., R.D. Tankersley Jr, J.K. Fiedler, and N.S. Nicholas, Assessment and prediction of bird and bat mortality at wind energy facilities in the Southeastern United States, in *Final Report Tennessee Valley Authority Knoxville TN*. 2005.
  10. Kjetil, B., Biological and conservation aspects of bird mortality caused by electricity power lines: a review. *Biological Conservation*, 1998. 86(1): p. 67-76.
  11. King, A.S. and J. McLelland, *Birds : their structure and function*. 1984, London; Philadelphia: Baillière Tindall.
  12. Catchpole, C. and P.J.B. Slater, *Bird song : biological themes and variations*. 2008, Cambridge [England]; New York: Cambridge University Press.
  13. Hackett, S.J., R.T. Kimball, S. Reddy, R.C.K. Bowie, E.L. Braun, M.J. Braun, J.L. Chojnowski, W.A. Cox, K.-L. Han, J. Harshman, C.J. Huddleston, B.D. Marks, K.J. Miglia, W.S. Moore, F.H. Sheldon, D.W. Steadman, C.C. Witt, and T. Yuri, A phylogenomic study of birds reveals their evolutionary history. *Science*, 2008. 320: p. 1763-1768.
  14. Ball, S.C., Fall bird migration of the Gaspe Peninsula. *Peabody Mus Nat Hist Yale Univ Bull*, 1952. 7: p. 1-211.
  15. Evans, W.R. and D.K. Mellinger, Monitoring grassland birds in nocturnal migration. *Studies in Avian Biology*, 1999. 19: p. 219-229.
  16. Mills, H., Automatic detection and classification of nocturnal migrant bird calls. *Journal of the Acoustical Society of America*, 1995. 97: p. 3370.
  17. Andersson, T., *Audio Classification and Content Description, in Technology*. 2004.
  18. Nillson, M. and M. Ejnarsson, *Speech Recognition Using Hidden Markov Model*,

- performance evaluation in noisy environment, in Department of Telecommunications. 2002, Blekinge Institute of Technology
19. Liu, J., J. Sun, and S. Wang, Pattern Recognition : An overview. Journal of Computer Science, 2006. 6: p. 57-61.
  20. Fagerlund, S., Automatic Recognition of Bird Species by Their Sounds. Science, 2004.
  21. Guyon, I. and A. Elisseeff, An introduction to feature extraction, in Analysis. 2006, Springer Verlag. p. 1-25.
  22. Fagerlund, S., Bird Species Recognition Using Support Vector Machines. EURASIP Journal on Advances in Signal Processing, 2007.
  23. Kogan, J.A. and D. Margoliash, Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study. The Journal of the Acoustical Society of America, 1998. 103(4): p. 2185-2196.
  24. Vaca-Castaño, G. and D. Rodriguez. Using syllabic Mel cepstrum features and k-nearest neighbors to identify anurans and birds species. in Signal Processing Systems SIPS 2010 IEEE Workshop on. 2010.
  25. Selin, A., J. Turunen, and J.T. Tanntu, Wavelets in recognition of bird sounds. EURASIP J. Appl. Signal Process., 2007(1): p. 141-141.
  26. Verma, G.K., Multi-feature Fusion for Closed Set Text Independent Speaker Identification
- Information Intelligence, Systems, Technology and Management, S. Dua, S. Sahni, and D.P. Goyal, Editors. 2011, Springer Berlin Heidelberg. p. 170-179.

27. Schrama, T., M. Poot, M. Robb, and H. Slabbekoorn, Automated recording, detection and identification of nocturnal flight calls: Results of a pilot study during autumn migration in the Netherlands. *Journal of Ornithology*, 2006. 147(5): p. 248-248.
28. Placer, J. and C.N. Slobodchikoff, A fuzzy-neural system for identification of species-specific alarm calls of Gunnison's prairie dogs. *Behavioural Processes*, 2000. 52: p. 1-9.
29. Phelps, S.M. and M.J. Ryan, Neural networks predict response biases of female tungara frogs. *Proceedings of the Royal Society of London Series B*, 1998. 265: p. 279-285.
30. Deecke, V.B., J.K. Ford, and P. Spong, Quantifying complex patterns of bioacoustic variation: use of a neural network to compare killer whale (*Orcinus orca*) dialects. *Journal of the Acoustical Society of America*, 1999. 105: p. 2499-2507.
31. Mirzaei, G., M.W. Majid, M.M. Jamali, Jeremy Ross, J. Frizado, P.V. Gorsevski, and V. Bingman, The application of Evolutionary Neural Network for bat echolocation calls recognition, in *Neural Networks (IJCNN), The 2011 International Joint Conference on 2011*. p. 1106-1111.
32. International Bioacoustics Council (IBAC).
33. Au, W.W.L., Animal bioacoustics. *Journal of the Acoustical Society of America*, 1999. 106: p. 1204.
34. Pavan, B.G. and U. Pavia, Short field course on bioacoustics *Bioacoustics Underwater Bioacoustics Soundscapes Bioacoustics for Taxonomy. Bioacoustics The International Journal Of Animal Sound And Its Recording*, 2008: p. 1-15.
35. Anderson, P., Monitoring Ultrasound, in *Monitoring Times*. 2009. p. 68.

36. Okanoya, K., Avian Bioacoustics  
Handbook of Signal Processing in Acoustics, D. Havelock, S. Kuwano, and M. Vorländer, Editors. 2009, Springer New York. p. 1887-1895.
37. Ballou, G., Handbook for Sound Engineers: The New Audio Cyclopedia edited by Glen Ballou, in Journal of the Acoustical Society of America. 1988. p. 2466.
38. Oppenheim, A.V., Applications of digital signal processing. Measurement, 1978. 1: p. 213-223.
39. Proakis, J.G. and D.G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Digital Signal Processing. Vol. 2nd ed. 1996: Prentice Hall. 471.
40. Lewis, B., ed. Bioacoustics: A Comparative Approach. 1983.
41. Song Scope: Bioacoustics Software Version 3.3 Documentation.
42. Raven Manual Appendix A: Digital Representation of Sound Available from: Digital Representation of Sound.
43. Oppenheim, A.V. and A.S. Willsky, Signals and Systems. October, 2003: p. 957.
44. Raven Manual Appendix B: A Biologist's Introduction to Spectrum Analysis.
45. Frigo, M. and S.G. Johnson, The Design and Implementation of FFTW3. Proceedings of the IEEE, 2005. 93: p. 216-231.
46. Krawitz, R.L. and S.L. Johnsson, Cooley-Tukey FFT on the connection machine. Parallel Computing, 1992.
47. Rapuano, S. and F.J. Harris, An introduction to FFT and time domain windows. IEEE Instrumentation Measurement Magazine, 2007. 10: p. 32-44.
48. Lyons, R., Windowing functions improve FFT results [spectral analysis]. Test Measurement World, 1998. 18: p. 37-38.

49. Baker, M.C. and D.M. Logue, Population Differentiation in a Complex Bird Sound: A Comparison of Three Bioacoustical Analysis Procedures. *Ethology*, 2003. 109(3): p. 223-242.
50. Mellinger, D.K. and P.M.E. Laboratory, Ishmael: 1.0 user's guide ; Ishmael : integrated system for holistic multi-channel acoustic exploration and localization. 2002: NOAA, Pacific Marine Environmental Laboratory.
51. Liu, R.C., K.D. Miller, M.M. Merzenich, and C.E. Schreiner, Acoustic variability and distinguishability among mouse ultrasound vocalizations. *The Journal of the Acoustical Society of America*, 2003. 114(6): p. 3412-3422.
52. F Boll, S., Suppression of acoustic noise in speech using spectral subtraction. *Acoustics Speech and Signal Processing IEEE Transactions on*, 1979. 27(2): p. 113-120.
53. McCallum, A., Birding by ear, visually. Part 1: Birding acoustics, in *Birding*. 2010. p. 50-63.
54. Bird Calls. Available from: <http://birds.ecoport.org/Behaviour/EBcall.htm>.
55. Evans, W.R. and K.V. Rosenberg, Acoustic Monitoring of Night-Migrating Birds : A Progress Report. New York, 2005: p. 1-17.
56. Marler, P., Bird calls: their potential for behavioral neurobiology. *Annals Of The New York Academy Of Sciences*, 2004. 1016: p. 31-44.
57. Flight Calls of Migratory Birds Available from: <http://oldbird.org/fcmbirds.htm>.
58. RAVEN Software. Available from: <http://www.birds.cornell.edu/brp/raven/RavenOverview.html>.
59. Sound Analysis Pro Software. Available from:

- [http://ofer.sci.ccny.cuny.edu/sound\\_analysis\\_pro](http://ofer.sci.ccny.cuny.edu/sound_analysis_pro).
60. Acoustic Monitoring: SM2 Night Flight Call Package. Available from:  
<http://www.wildlifeacoustics.com/products/acoustic-monitoring>.
61. Wildlife Acoustics. Available from: <http://www.wildlifeacoustics.com/>.
62. Ohio Spring Migration. Available from: <http://www.birdnature.com/spoh.html>.
63. Spring Migration in the Lake Erie Marsh Region. Available from:  
[http://www.bsbo.org/passerine/migration\\_timetable.htm](http://www.bsbo.org/passerine/migration_timetable.htm).
64. Bankman, I.N., Handbook of Medical Image Processing and Analysis. SPIE Press,  
ed. M. Sonka and J.M. Fitzpatrick. Vol. 1. 2009: Elsevier. 725-845.
65. Li, D., I.K. Sethi, N. Dimitrova, and T. McGee, Classification of general audio data  
for content-based retrieval. Pattern Recogn. Lett., 2001. 22(5): p. 533-544.
66. Boggess, A., F.J. Narcowich, D.L. Donoho, and P.L. Donoho, A First Course in  
Wavelets with Fourier Analysis, in Physics Today. 2002, Wiley. p. 63.
67. Polikar, R. The Wavelet Tutorial. Available from:  
<http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>.
68. Mallat, S., A Wavelet Tour of Signal Processing: The Sparse Way. Book. 2009:  
Academic Press. 805.
69. Mallat, S.G., A Theory for Multiresolution Signal Decomposition. IEEE Transactions  
on Pattern Analysis and Machine Intelligence, 1989. 11: p. 674-693.
70. Soman, K.P., K.I. Ramachandran, and N.G. Resmi, Insight Into Wavelets From  
Theory To Practice. 2004, New Delhi: Prentice Hall of India.
71. Edwards, T., Discrete Wavelet Transforms : Theory and Implementation Tim  
Edwards ( [tim@sinh.stanford.edu](mailto:tim@sinh.stanford.edu) ). Universidad de, 1992: p. 1-26.

72. Morlet, J., Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. *Geophysics*, 1982. 47: p. 203.
73. Morlet, J., G. Arens, and E. Fourgeau, Wave-Propagation and Sampling Theory 2. Sampling Theory and Complex Waves. *Geophysics* Vol 47 Issue 2 pp 222236, 1982.
74. Hasan, R., M. Jamil, G. Rabbani, and S. Rahman, Speaker identification using mel frequency cepstral coefficients. *Time*, 2004: p. 28-30.
75. Sigurdson, S., K.B. Petersen, and J. Larsen, Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music. *Victoria*, 2006: p. 3-6.
76. Hill, P.C.J., Dennis Gabor - Contributions to Communication Theory & Signal Processing. *EUROCON 2007 The International Conference on Computer as a Tool*, 2007: p. 2632-2637.
77. Deng, L. and D. O'Shaughnessy, Discrete-Time Signals, Systems and Transforms, in *Speech processing: a dynamic and optimization-oriented approach*. 2003, CRC Press: New York. p. 15-67.
78. Allen, J.B. and L.R. Rabiner, A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, 1977. 65(11): p. 1558-1564.
79. Maragos, P., A Representation Theory for Morphological Image and Signal Processing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1989. 11(6): p. 586-599.
80. Van Den Boomgaard, R. and A. Smeulders, The morphological structure of images: the differential equations of morphological scale-space. *Pattern Analysis and Machine Intelligence IEEE Transactions on*, 1994. 16: p. 1101-1113.
81. Dougherty, E.R., An Introduction to Morphological Image Processing. *Tutorial Texts in Optical Engineering*. Vol. TT9. 1992: A publication of SPIE - the Int. Society for



- Optical Engineering (SPIE Press). 161.
82. Dougherty, E.R. and A. Introduction, Foundations of morphological image processing  
14. Analysis, 1999. 6: p. 2722-2722.
  83. Alpaydin, E., Introduction to machine learning / Ethem Alpaydin. 2004, Cambridge, Mass. : MIT Press. xxx, 415 p. :.
  84. Rabiner, L.R., Rabiner 1989 Tutorial on HMM and selected applications.pdf. Proceedings of the IEEE, 1989. 77(2): p. 257-286.
  85. Munakata, T., Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More. Language. 2008: Springer.
  86. ROSENBLATT, F., The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 1958. 65: p. 386-408.
  87. Block, H.D., The Perceptron: A Model for Brain Functioning. Reviews of Modern Physics, 1962. 34: p. 123-135.
  88. Holland, J.H., Adaptation in Natural and Artificial Systems. Ann Arbor MI University of Michigan Press. Vol. Ann Arbor. 1975: University of Michigan Press. 211.
  89. Man, K.F., K.S. Tang, and S. Kwong, Genetic algorithms: concepts and applications [in engineering design]. IEEE Transactions on Industrial Electronics, 1996. 43: p. 519-534.
  90. Blanco, A., M. Delgado, and M.C. Pegalajar, A genetic algorithm to obtain the optimal recurrent neural network. International Journal of Approximate Reasoning, 2000. 23(1): p. 67-83.
  91. Chen, M. and Z. Yao, Classification Techniques of Neural Networks Using Improved

Genetic Algorithms. 2008 Second International Conference on Genetic and Evolutionary Computing, 2008: p. 115-119.

# Appendix A

## Other Experiments Performance

### Hypothetical Bird Counting

If it can be assumed that, from the first detectable bird call, the x, y, and z coordinates of the bird can be precisely obtained using the microphone array, the bird is traveling at a constant speed, and that the bird is flying linearly, then the information can be known about the location of that bird when it makes its next detectable call. This information is represented in Figure A.1 as a series of points, which make up a sphere, where that bird could potentially be.

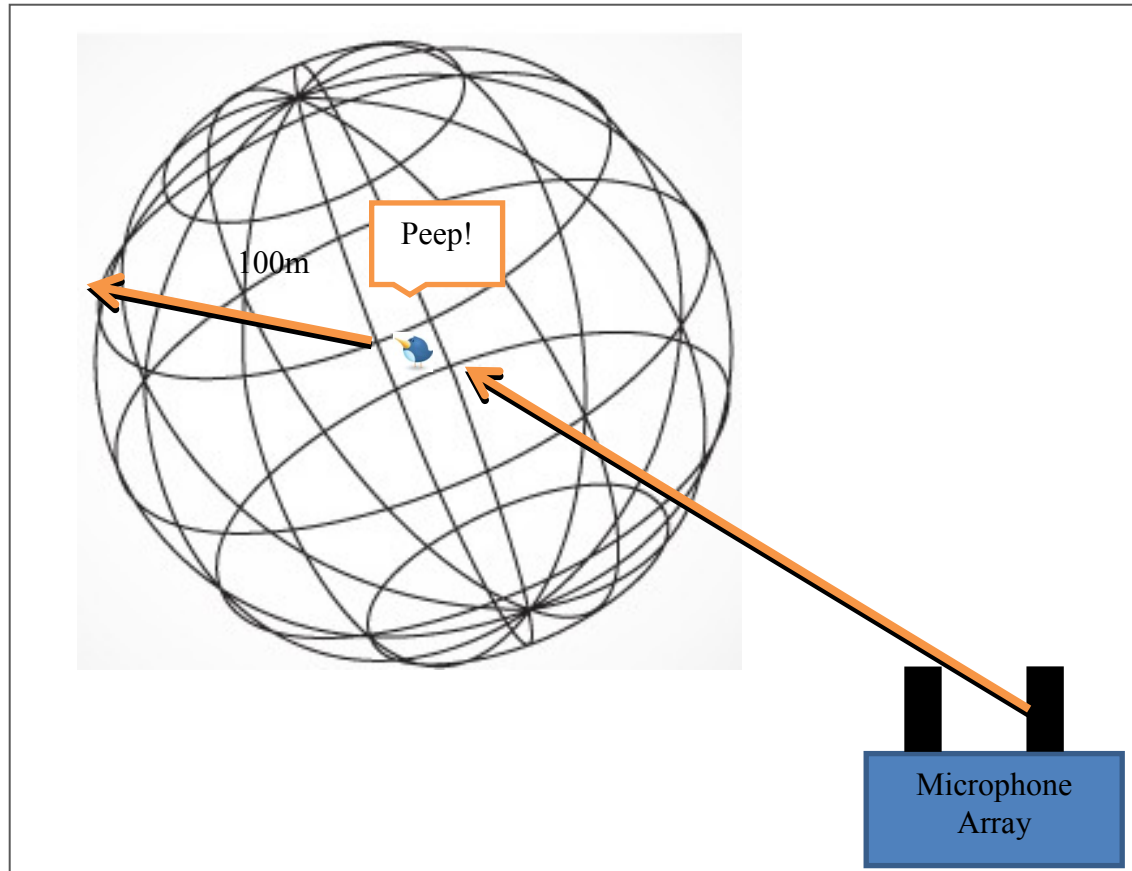


Figure A-1 Representation of hypothetical bird counting method

This information would be useful when attempting to obtain an accurate bird count in an environment where there are more than one bird of the same species in the range of the microphone array at one time. If a second call is made, and the location of the origin on that call does not lie on the sphere of points, then it can be said that it is coming from a separate bird. Since this technique requires the relation of one call to a previous one, a restriction governing the length of time between points that are to be related to one another, depending on the recording distance and bird speed, must be implemented.

In this simplified system, it was assumed that the “birds” flight path was linear and unidirectional and that the bird’s call always has the same volume. Since this was only a

test, the “bird” call was a distinct sound which was recorded at various known distances from the microphone. These recordings were analyzed and the amplitude of the “bird” call was found using Song Scope. Then, an amplitude versus distance graph was plotted, which can be seen in Figure A.2, using Microsoft Excel and a function was found. Using this function, the location of the “bird,” from its initial and subsequent call, could be determined in this much simpler situation.

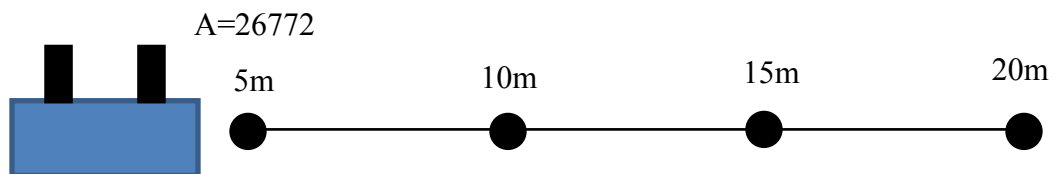
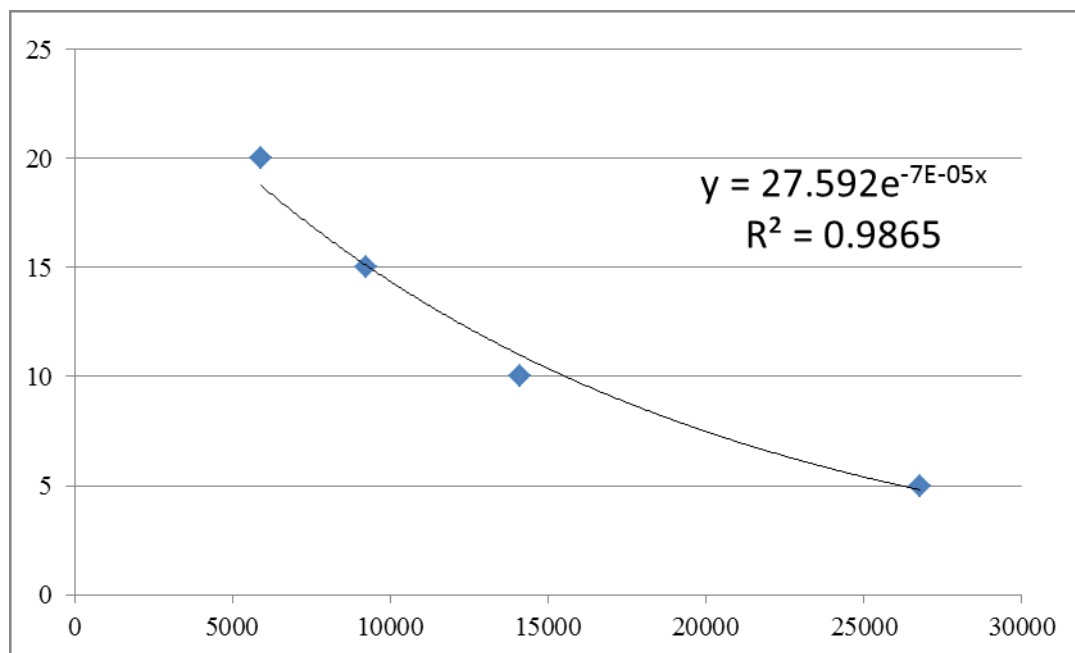


Figure A-2: Amplitude vs. distance graph for hypothetical bird call counting

As a simple demonstration of how this technique could work, Figure A.3 was constructed.

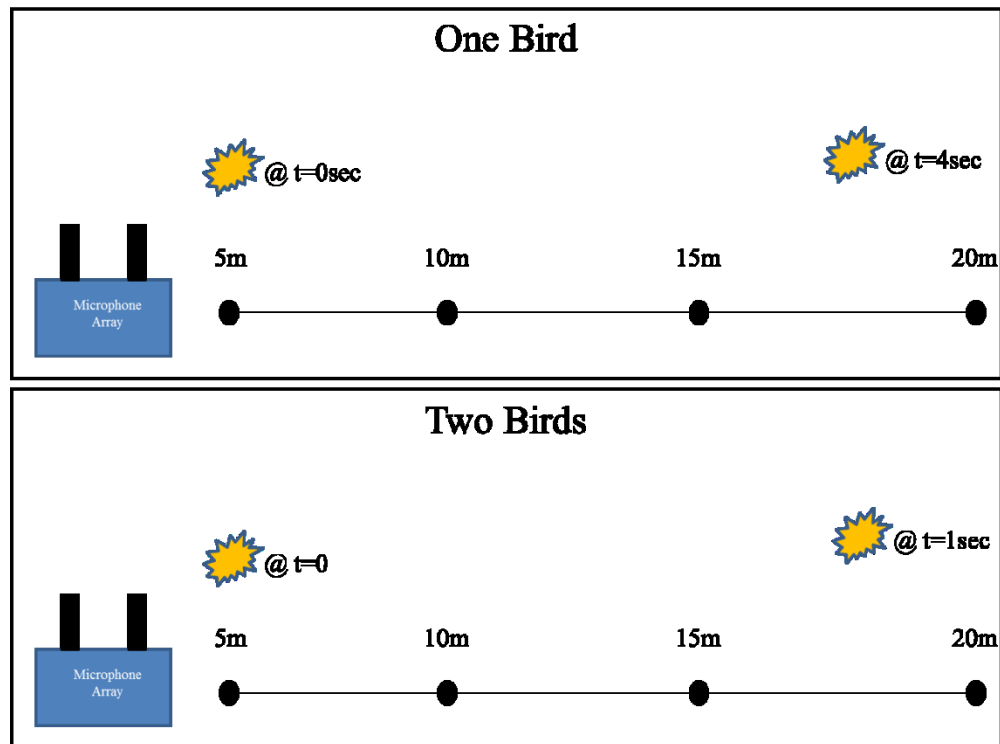


Figure A-3: Simple demonstration of hypothetical bird counting method

If one assumes that a bird, with the speed of 3 m/s, has an initial call that is detected, at  $t=0\text{sec}$ , 5m away from the microphone and another call that is detected, at  $t=4\text{sec}$ , 17m away from the microphone, then it is assumed these two calls are coming from the same bird. This is because the bird will fly 12m further from the microphone in this 4 second time span between the calls. On the other hand, if two calls were detected in the same location but only 1 second apart, then it would be assumed that they were coming from two separate birds.