12-2014

Speaker Dependent Voice Recognition with Word-Tense Association and Part-of-Speech Tagging

Theodore Philip King Ernst

SPEAKER DEPENDENT VOICE RECOGNITION WITH WORD-TENSE ASSOCIATION AND PART-OF-SPEECH TAGGING

By

Theodore Philip King Ernst

A Thesis Submitted to the College of Engineering Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

December 2014

SPEAKER DEPENDENT VOICE RECOGNITION WITH WORD-TENSE ASSOCIATION AND PART-OF-SPEECH TAGGING

by

Theodore Philip King Ernst

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Eric J. Coyle, Professor, Daytona Beach Campus, and Thesis Committee Members Dr. Patrick Currier, Professor, Daytona Beach Campus, and Dr. Yan Tang, Professor, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Thesis Review Committee:

Eric J. Coyle
Committee Chair

Patrick Currier
Committee Member

Yan Tang
Committee Member

Jean-Michel Dhainaut, Ph.D.
Graduate Program Chair,
Mechanical Engineering

Charles F. Reinholtz, Ph.D
Department Chair,
Mechanical Engineering

Maj Mirmirani, Ph.D.
Dean, College of Engineering

Robert Oxley, Ph.D.
Associate Vice President of Academics

12-9-14
Date

Acknowledgements:

I would like to thank everyone whose help, support and encouragement has helped me through the completion of this thesis.

Firstly, I would like to thank God, who gave me the strength to sit up nights and the knowledge to complete this project without which I would not be at the position I am today.

I extend my sincere gratitude to my mentor and thesis committee chair, Dr. Eric J. Coyle, who has an abundance of patience. He never lost faith in my thesis when I faltered and got stuck. He gave me the jolt I needed to continue my thesis with vigor and renewed determination. His continued guidance has taught me a valuable lesson regarding research methods.

I thank my thesis committee members, Dr. Patrick Currier and Dr. Yan Tang, who accepted to be on the thesis committee. I would like to thank them for their continued interest in this thesis and their co-operation with scheduling the thesis defense.

I would like to thank the students in Dr. Eric Coyle's thesis group and my roommates that took the time and effort to provide me with the audio data and testing trials, which I required for the Speaker recognition system.

Finally, but by no means the least, I would like to thank my family (esp. Mom, Jo & Carol) for their continued prayers and support through this entire period of study. I dedicate this thesis to them and their faith in me.

Abstract:

Extensive Research has been conducted on speech recognition and Speaker Recognition over the past few decades. Speaker recognition deals with identifying the speaker from multiple speakers and the ability to filter out the voice of an individual from the background for computational understanding. The more commonly researched method, speech recognition, deals only with computational linguistics. This thesis deals with speaker recognition and natural language processing. The most common speaker recognition systems are Text-Dependent and identify the speaker after a key word/phrase is uttered. This thesis presents Text-Independent Speaker recognition systems that incorporate the collaborative effort and research of noise-filtering, Speech Segmentation, Feature extraction, speaker verification and finally, Partial Language Modelling. The filtering process was accomplished using 4^{th} order Butterworth Band-pass filters to dampen ambient noise outside normal speech frequencies of 300Hzto3000Hz. Speech segmentation utilizes Hamming windows to segment the speech, after which speech detection occurs by calculating the Short time Energy and Zero-crossing rates over a particular time period and identifying voiced from unvoiced using a threshold. Audio data collected from different people is run consecutively through a Speaker Training and Recognition Algorithm which uses neural networks to create a training group and target group for the recognition process. The output of the segmentation module is then processed by the neural network to recognize the speaker. Though not implemented here due to database and computational requirements, the last module suggests a new model for the Part of Speech tagging process that involves a combination of Artificial Neural Networks (ANN) and Hidden Markov Models (HMM) in a series configuration to

achieve higher accuracy. This differs from existing research by diverging from the usual single model approach or the creation of hybrid ANN and HMM models

Table of Contents:

List of Figures:

List of Tables:

Definitions or Nomenclature:

1) Voice Biometrics – Voice Biometrics allow for the control of access to information or characteristics based on a person's voice print. It deals with calculations of pitch, amplitude, rate of speech, etc.
2) Signal Domain Representations – A signal can be represented in the time domain and frequency domain. In the time domain, the signal is represented as amplitude over a period of time. In the frequency domain, the signal is represented as the magnitude and phase of the signal at each frequency.
3) Noise
    a. Additive noise: It is a basic noise model used in information theory to mimic the effect of a random process that occurs in nature.
    b. Multiplicative Noise: This refers to an unwanted random signal that gets multiplied by the actual signal during capture or recognition.
4) SNR: This is a measure that compares the level of a desired signal to the level of background noise. It the ratio of signal power to noise power.
5) Robustness: Ability of a system to resist change without adapting its initial state configuration.
6) Finite Impulse Response filter (FIR filter): This filter is said to have finite response because it settles to zero in a finite time.
7) Feed Forward neural Network: Artificial Neural network where the connections between the units do not form a directed cycle. Information moves only in one direction from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network.

8) Radial Basis Neural Network: Artificial neural network that uses radial basis functions as the activation function. Output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.
9) Finite State Grammar: Finite state grammar is a database of generative/transformational grammar that states all possible transformations and states defined by the grammatical constraints of the language.

1. Introduction:

1.1. Voice Recognition

Voice Recognition can be divided into two major parts, namely Speech Recognition and Speaker Recognition. Speaker Recognition has been studied for over four decades and uses the acoustic features of speech that have been found to differ between individuals such as pitch, rate of speech, frequency range, etc.[1]"Speaker Recognition is defined as the process of identifying the person, by inferring speaker-specific characteristics *(voice biometrics)* from the voice of the individual." This is different from Speech Recognition which involves translating speech into text, irrespective of the speaker. Unlike speech recognition, where apriori training is impractical due to the large number of target linguistics, speaker recognition can utilize apriori training of key works or voice biometrics, resulting in a more robust speech recognition algorithm. In previous work, this has been achieved through various methods including Hidden Markov models (HMM) [17, 18], and Artificial Neural Networks (ANN) [15, 17 & 18].

[2] The task of Speaker recognition has been classified based on the application and level of automation [5, 11, 12, & 14]. For example, Speaker recognition for basic tasks such as initial identification requires that the speaker only utter a certain set of words in the beginning. This eliminates the need for continuous speaker identification. For other applications that require continuous security, the system should be able to identify the speaker throughout the process and accept inputs only if the authorized speaker is talking. Thus, Speaker Recognition has been divided into two subclasses, namely, Text-Dependent and Text-Independent Speaker Recognition. Text-Dependent

speaker recognition is one in which a particular set of words are used to create a test group. The system identifies the speaker when these set of words or a subset of the words is spoken. Text-Independent Speaker Recognition requires no such particular set of words to be spoken and derives the voice biometrics (pitch, zero-correlation rate, etc.) from any set of words and associates those characteristics to the speaker. Ref. [2] further explains the fundamental differences between Text-Dependent and Independent Speaker Recognition by dividing signal processing into certain key steps listed below. These steps outline the requirements for basic signal processing such as tuning a signal to have low signal to noise ratio and filtering out unwanted frequencies that could affect the signals features/characteristics. It further outlines the basic procedure for speech processing such as classifying the input signal into its characteristics knows as feature vectors and tuning those vectors to associate with certain speech patterns stored during the system training phase.

The basic system architecture for Speaker Recognition as defined in [2] consists of the following elements and organized as indicated in Figure 1-1, Noise Filtration, Speech Segmentation, Feature Extraction, Pattern Matching/Association, and Pattern Comparison.

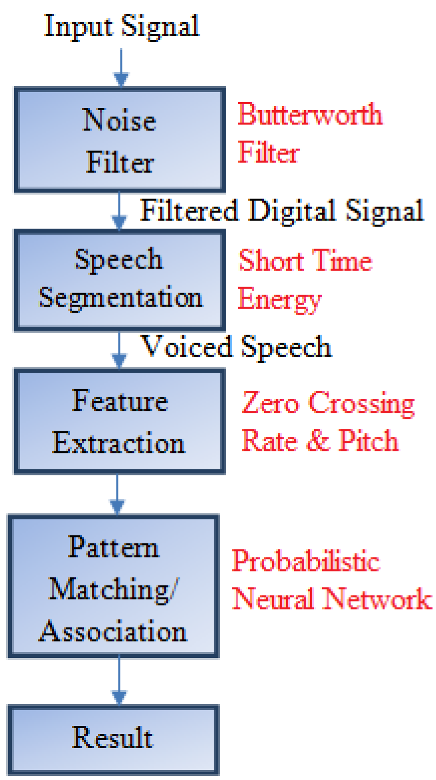


Figure 1-1: Basic System Architecture for Speaker Recognition

This thesis also addresses the use of Natural Language Processing (NLP) in coordination with Speaker Recognition (SR). These two modules, SR and NLP, work as explained in sections 1.2 and section 1.3, as illustrated in figure 1-2.

1.2 Natural Language processing

[25]Natural Language modeling/processing (NLP) is a field of computer science, artificial intelligence and linguistics. Statistical methods and Probabilistic methods were the first methods used in the coding of NLP algorithms and involved integrating a large set of rules which governed the selection process of the algorithm in the various tasks listed in Table 5-1. This proved to be time consuming and required detailed knowledge on every possible grammar constraint related to the particular language the system was being trained to process. Modern day methods of NLP get rid of this problem by involving machine learning algorithms to learn such rules based on the analysis of a knowledge base of real world examples.

Existing research on Natural Language processing has shown that the most popular methods involved in language processing are Hidden Markov models [35], [36], and Neural networks [37], [38]. Hidden Markov models have been researched for this purpose since the mid 1980's, with the first database being the CLAWS database started at that time and created by Garside and Leech with an accuracy of 96-98%. Research in this field of HMM has always dealt with different approximation algorithms such as the Viterbi approximation, the forward-backward and maximum likelihood probabilities to solve the problem of tagging words. The advantage to this method is that it allows for the probability to transition from one tag to the next based on the previous word tag. Whereas, Neural networks deal with the tagging process as a system that trains based on previous databases and calculates the transition probability based on natural conversations databases.

1.3 Relation to existing research

While existing research proposes methods of implementing each Speaker Recognition element, previous research fails to properly address the collective participation of these elements with the addition of Language modeling. Also, previous research [2] comprises of recognition of a single speaker with large speech samples required for training (ref [14], [15], [16], [17] & [18]). This thesis is aimed at recognizing multiple speakers in quick succession with single paragraph training consisting of 50-100 words. This is important as it saves time and effort on the part of the user to train and initiate the system, while maintaining accuracy. Also, Natural language processing in Section 4.3, although not implemented, proposes a concept utilizing artificial neural networks and Hidden Markov Model in a series configuration to improve accuracy of the Language processing module. Most current speech recognition tools have less accuracy due to small databases or large computation times due to large databases. This method as explained in Section 5 speculates accuracy improvement of the language module which can help in text prediction and text correction by gauging the meaning and context of the text.

Section 2 explains the importance of noise filtering to attain a signal with a low signal to noise ratio and previously proposed methods for implementing this process. This paper implements the Butterworth filter versus the Chebyshev filter and lists the results in section 2.3. Section 3 explains the importance of segmenting the speech into parts containing voiced & unvoiced. It further lists previously introduced methods such as Zero crossing rate and Short time energy intervals. Section 3 explains why energy segmentation was chosen and presents the results of this implementation. Section 4 consists of steps 3 - 5 of the system architecture mentioned in section 1.1. It divides the core process of Speaker recognition into data collection and final recognition. It explains formulae used for extracting various characteristics (pitch, zero crossing, etc.) and the Radial Bias Probabilistic Neural network implemented to recognize the speaker. Section 5 outlines the final step of this thesis which involves utilizing predefined functions and methods to translate the speech to text and then running the speech through an Artificial neural network and Hidden Markov model in a series configuration which identifies the tense of the word and the part of speech to which the word is associated. This process was not implemented due to an insufficient database and only speculates an improved accuracy for the tagging process.

The complete architecture for this system is shown below in Figure 1-2.

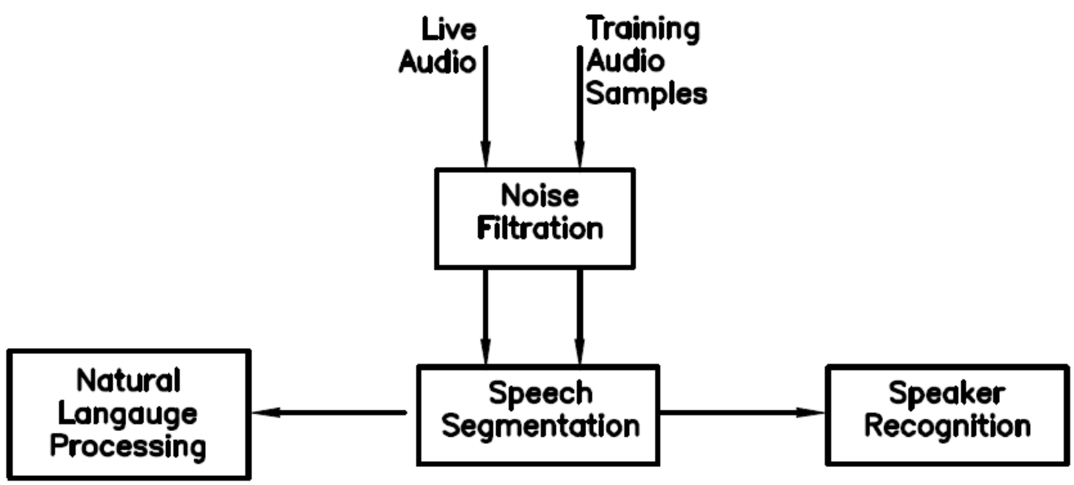


Figure 1-2: System Architecture incorporating both SR and NLP.

Note: - The system architecture above illustrates the architecture that was utilized in this thesis. It is an extension of the architecture illustrated in Figure 1.1and shows that the audio from the Speech Segmentation segment is utilized for both SR and NLP modules.

1.4 Applications:-

Speaker recognition has multiple applications in today's world, the most popular of which is security. Consider that, in today's world, with the advent of social networking, information storage and retrieval and automation, these fields are plagued with the requirement of passwords and firewalls to safeguard privacy. Speaker recognition systems can make these safeguards more robust with the accuracy to recognize a person's voice only if that person is present. Another example for real world application would be towards certain systems that are required to only follow commands from a particular user, such as a robotic wheelchair. It would be required to listen only to the physically handicapped individual that the system is trained towards and ignore any command issued by an unknown user. It should also be trained to ignore noise or background sounds, so as to ensure efficient operation in noise environments. Another addition to speaker recognition would be to utilize the system to recognize the inflection in a person's voice and understand if the statement made by the user was a command or a statement made in general conversation. This would help it ignore statements that were not intended for the system in particular, but was made towards another person.

2 Noise Filtering:

2.1. Basics of digital Signal Processing and Filtration

[24]Digital Signal processing (DSP) is the control and manipulation of any signal to assess, quantify and modify the signal in terms of its characteristics. It can be represented in discrete time, continuous time, discrete frequency or other less common domains. Before and during conversion to digital format, the addition of noise to the signal is highly possible in the form of Additive noise, Multiplicative noise, etc. Any form of noise can largely affect any means of communication as it is present in the surrounding environment and changes the characteristics of the signal being processed increasing the signal-to-noise ratio (SNR). [2] In order for any system to function efficiently in the task of signal/speech processing and pattern verification, the digital signal with a high SNR has to be run through an algorithm that filters out the noise. This can be accomplished by assuming that the input signal is a superposition of the actual signal with noise. Thus, the objective becomes to restore the actual signal. The objective is more broadly classified in [2] by defining the performance criteria into the quality of the signal, level of noise, and the consistency of system output. Thus, Filter performance is gauged on preservation of the original signal in the passband and rejection of disturbances in the stopband.

In relation to the collective application of all modules that this thesis aims to show, the importance of noise filtration is that it makes all subsequent modules more effective. If successful, it preserves the voiced data needed to accurately represent the voice biometrics of the speaker. As the frequency of ambient noise ranges above and below the range of human speech, the true speech signal is distorted. As such, this can reduce the effectiveness of both SR and NLP.

[3] There are many classifications of filters, a number of which of which are useful in audio signal noise reduction such as Linear or non-linear filter, Discrete time or Continuous time filters and Infinite Impulse response or finite impulse response filters. For the purpose of this thesis, Infinite Impulse response (IIR) filters were originally considered, due to the fact that Finite Impulse response (FIR) systems require a higher order of filter to accomplish the minimal ripple requirement as well as more computation power. As seen in the work of McFarlane, Andy [6], the principle advantages of IIR filters are its minimal memory requirements, and computational efficiency when compared to its counterpart FIR filters. The principle drawback is that it is more sensitive to coefficient quantization errors. "Comparative study of different filters" by Abbas. S. Hassan [7], show a detailed comparison of Butterworth, Chebyshev and Elliptical filters with data results and the required orders to design a Low pass filter with various responses. In the works of Sorenson, Johannesen and others [5], studies were conducted to compare IIR filters in general to wavelet filters for reduction of Noise. It was found that the computation time was faster for IIR filters. An IIR filter is a filter whose impulse response does not settle to zero after a certain point, but continues on indefinitely. IIR filters can be based on well-known solutions for analog filters such as Chebyshev filter, Butterworth Filter and Elliptic Filter, inheriting the characteristics of those solutions. [4]For the purpose of the thesis, Butterworth and Chebyshev Infinite Impulse Response filters are considered due to their popularity in noise reduction tasks and because they

require smaller order of filter to implement for sufficient noise reduction. However, it was theorized that Butterworth filters are better than Chebyshev filters for the filtration process because they do not have ripples in the passband as will be shown in section 2.3 and they have low attenuation in the passband.

2.2. Butterworth Filter

An ideal filter is defined as a filter that has full transmission in the passband, complete attenuation in the stopband and an abrupt transition between the two bands. [8] Butterworth filters are different from filters such as Chebyshev filter and elliptic filter in the sense that it has minimal ripples or unwanted effects in the stopband or passband. . This is important because ripples are characterized as unwanted residual periodic variations in insertion loss with the frequency of the filter, which would distort speaker audio. Passband degradation of this nature is particularly destructive in speaker identification and speech recognition, where frequency content is used for identification. Stopband ripple also has an indirect effect on the signal by reducing the attenuation in the stopband and ultimately reducing the effectivity of the filter. Also, Chebyshev filters have a faster roll-off rate but create ripples in the stopband (Chebyshev 1 filter) or passband (Chebyshev 2 filter), whereas Butterworth filters have a more linear phase response in the passband with no ripples in the passband or stop band.

The Transfer function of a filter is most often defined in the domain of the complex frequencies. The transfer function *H(s)* is the ratio of the output signal *Y(s)* to the input signal *X(s)* as a function of the complex frequencies. [8]

$$H(s) = Y(s)/X(s)$$

With s = complex frequency.

The Gain of an n^{th} order low-pass Butterworth filter is given as

$$G^2(\omega) = |H(j\omega)|^2 = G_0^2 / (1 + (\omega/\omega_c)^{2n})$$

Where, n= order of filter,

ω_c*= cutoff frequency,*

G_0*= DC gain.*

From the above equation, we can approximate the transfer function for the Butterworth filter as,

$$H(s)= G_0^2 / (1 + (-s^2/ \omega^2{}_c)^n$$

Where, $s = j\omega$

The transfer function of the Chebyshev filter can be approximated as,

$$H(s) = \frac{1}{(2^{n-1})\mathcal{E}} * \prod_{m=1}^{n} 1/(s - s\text{^}(pm))$$

Where, s^(pm) are poles with negative sign in front of the real term.

2.3. Data and results

To verify which filter would be best suited for noise removal within a speaker recognition system, a male audio sample from the data collection discussed in section 4.1 was processed using multiple elliptical filters, Butterworth filters, and Chebyshev Filters. Elliptic filters have adjustable ripples in their passband. Varying the level of ripples changes the filters type to Chebyshev (1 or 2) or Butterworth filter. Since the reduction of noise in audio signal requires minimal ripples, elliptic filter was not chosen. The results and comparisons shown in Figure 2-1, 2-2, 2-3are of 2nd order Butterworth filter, Chebyshev1 filter and Chebyshev2 filters. They prove that Butterworth filters satisfy the requirement of no ripples with sufficient attenuation in the stopband.

<u>*Filter Frequencies:*</u>

Audio Sampling Frequency: 44100 Hz

Passband Frequency: 300 Hz to 3000 Hz (0.0136 to 0.1361 in range of 0-1)

Note: Passband frequencies were selected since the usable voice frequency bands ranges in telephony range from approximately 300 Hz to 3000 Hz [20]

As seen in Figure 2-1 and Figure 2-2, the Chebyshev 1 filter shows unwanted attenuation of the signal with a in the passband frequencies (and affects the characteristics of the voice). Thus, the Butterworth filter was selected for the purpose of noise reduction.

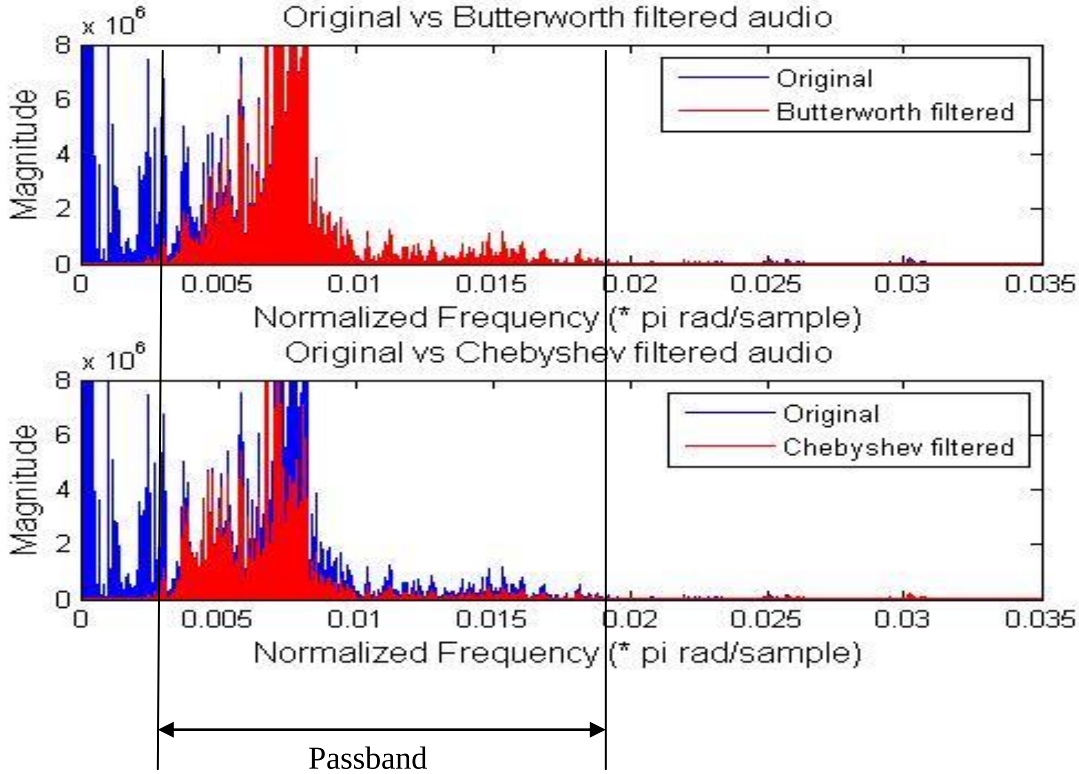


Figure 2-1: 2^{nd} Order Butterworth Filter vs. Chebyshev-1 Filter

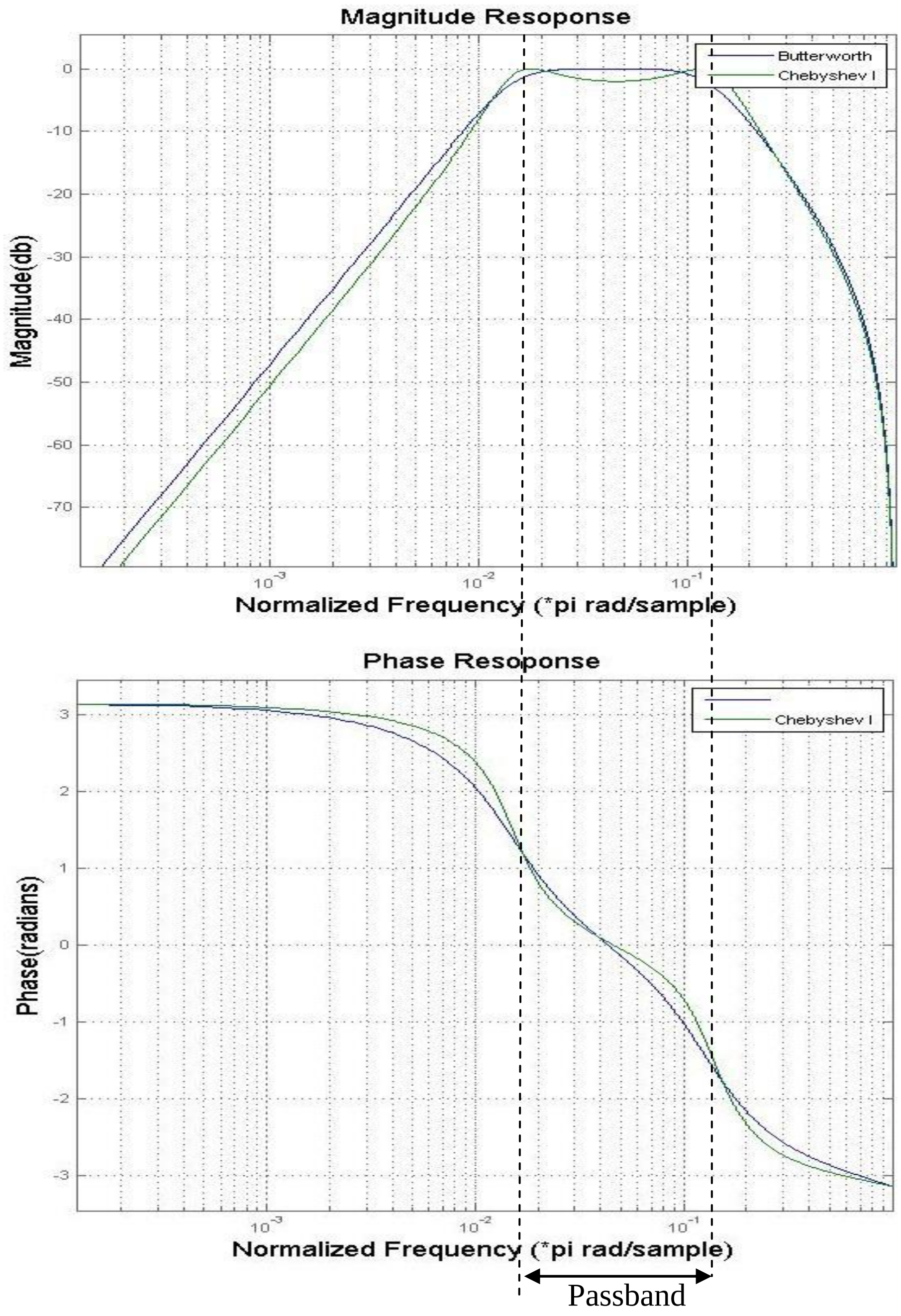


Figure 2-2: Magnitude and Phase response of second order Butterworth and Chebyshev-1 Filter

In Figure 2-3, the Chebyshev 2 Filter shows less attenuation in the stopband frequencies as well as a lot of noise before the passband which is an effect of the earlier mentioned ripple. When compared to the Butterworth Filter, the output of the Butterworth Filter does not show the ripple or as much attenuation in the passband. In Figure 2-4, we can see that the sudden change in the magnitude is actually a ripple that has disastrous effects on the audio represented in Figure 2-3. It creates a major change in the filter and is no longer valid for the purpose of Speaker Recognition.

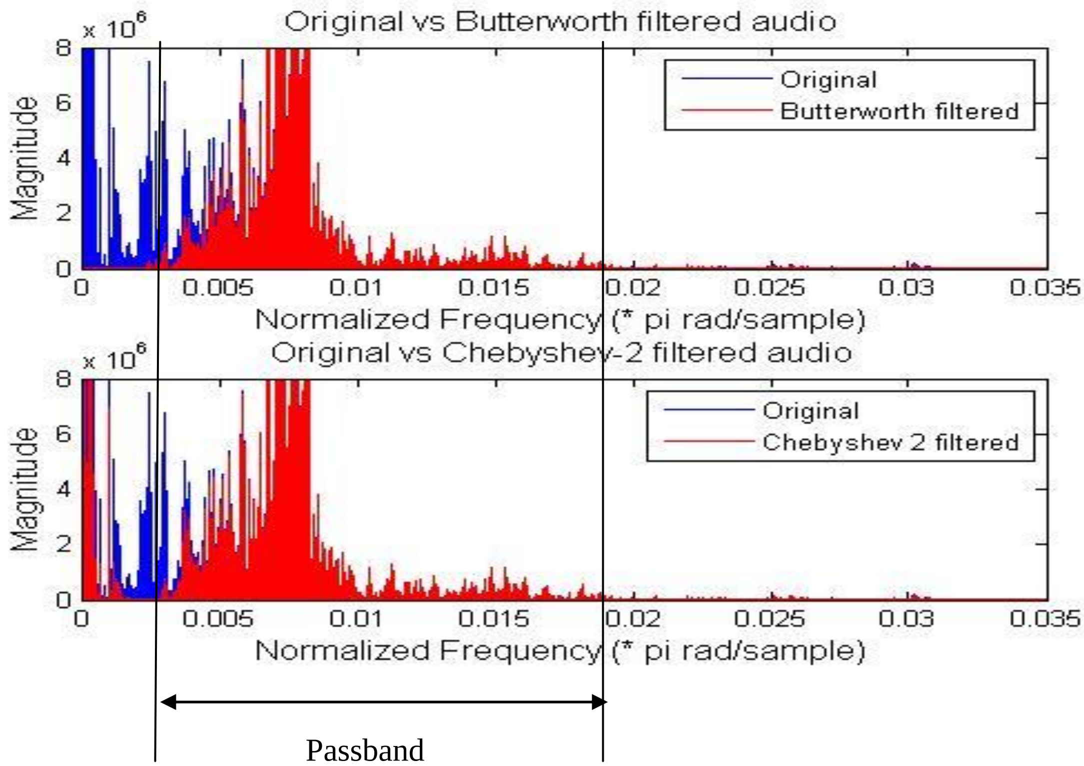


Figure 2-3:2nd Order Butterworth Filter vs. Chebyshev-2 Filter

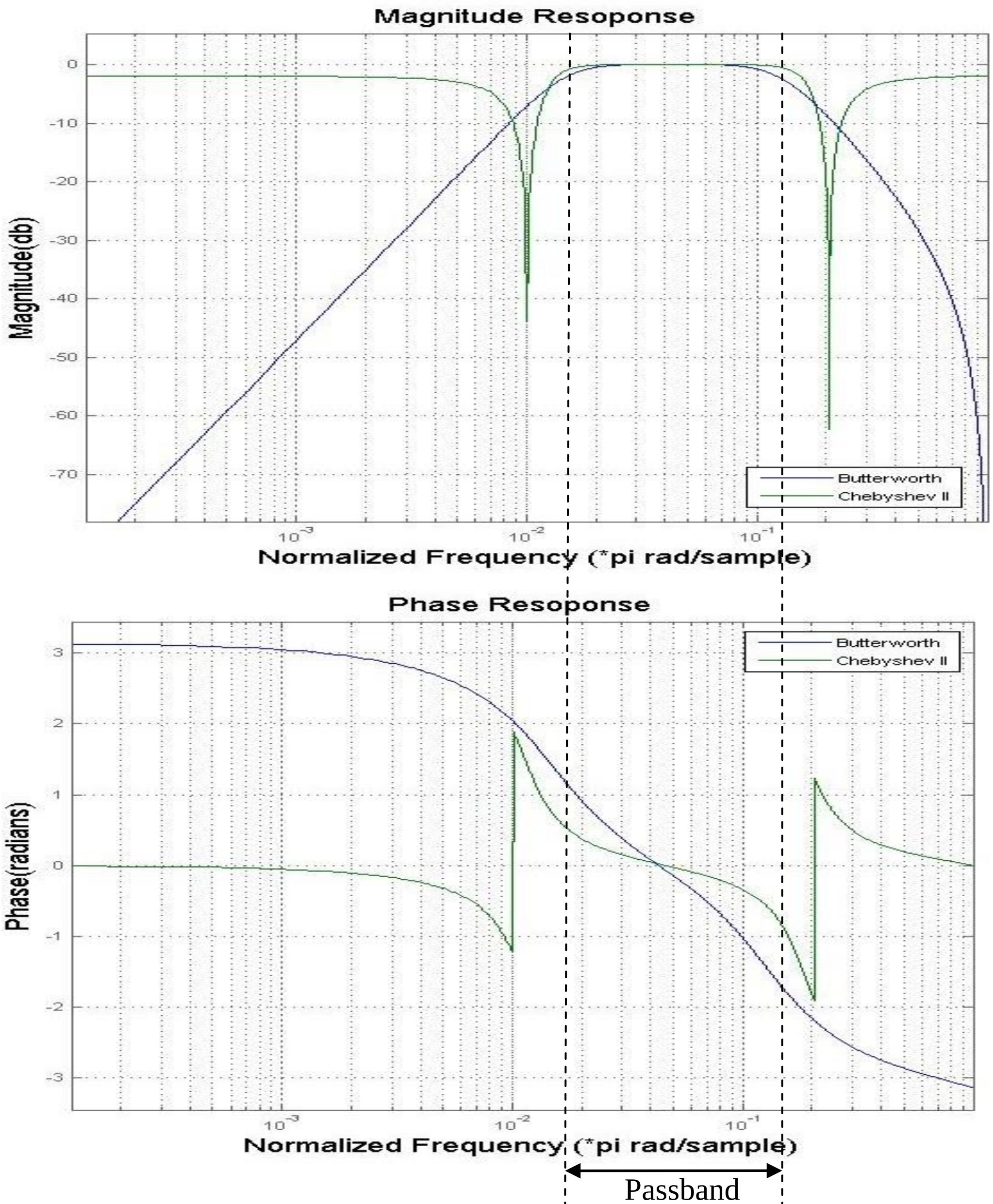


Figure 2-4: Magnitude and Phase response of second order Butterworth and Chebyshev-2 Filter

In terms of Butterworth Filters, a 4^{th} order filter was chosen for this project to achieve a higher attenuation in the stopband, faster roll off and create less than 1db of attenuation in the passband. Also, as seen in the figures below, the stopband had greater attenuation for the 4^{th} order filter than the stopband for the 2^{nd} order Butterworth filter presented. The 4^{th} order Butterworth filter performs better than the second order filter and Chebyshev type-I and type-II filters to give better stopband attenuation and lesser passband attenuation. Also, there are no ripples present in either passband or stopband. Thus the 4^{th} order Butterworth filter was used for this project as it serves the requirements of sufficient attenuation and minimal ripples required for Speech processing as mentioned in section 2.1. In case, another filter was to be chosen it would have to adhere to the same requirements.

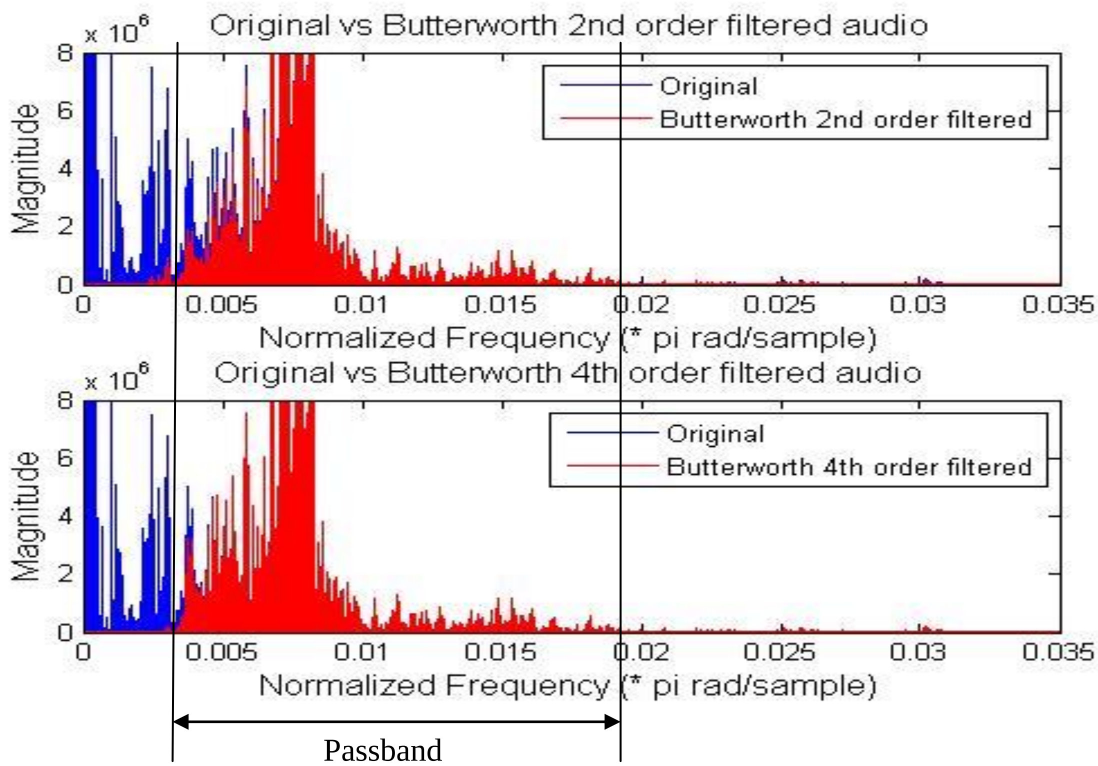


Figure 2-5: Butterworth 2nd order vs. 4th order Filter

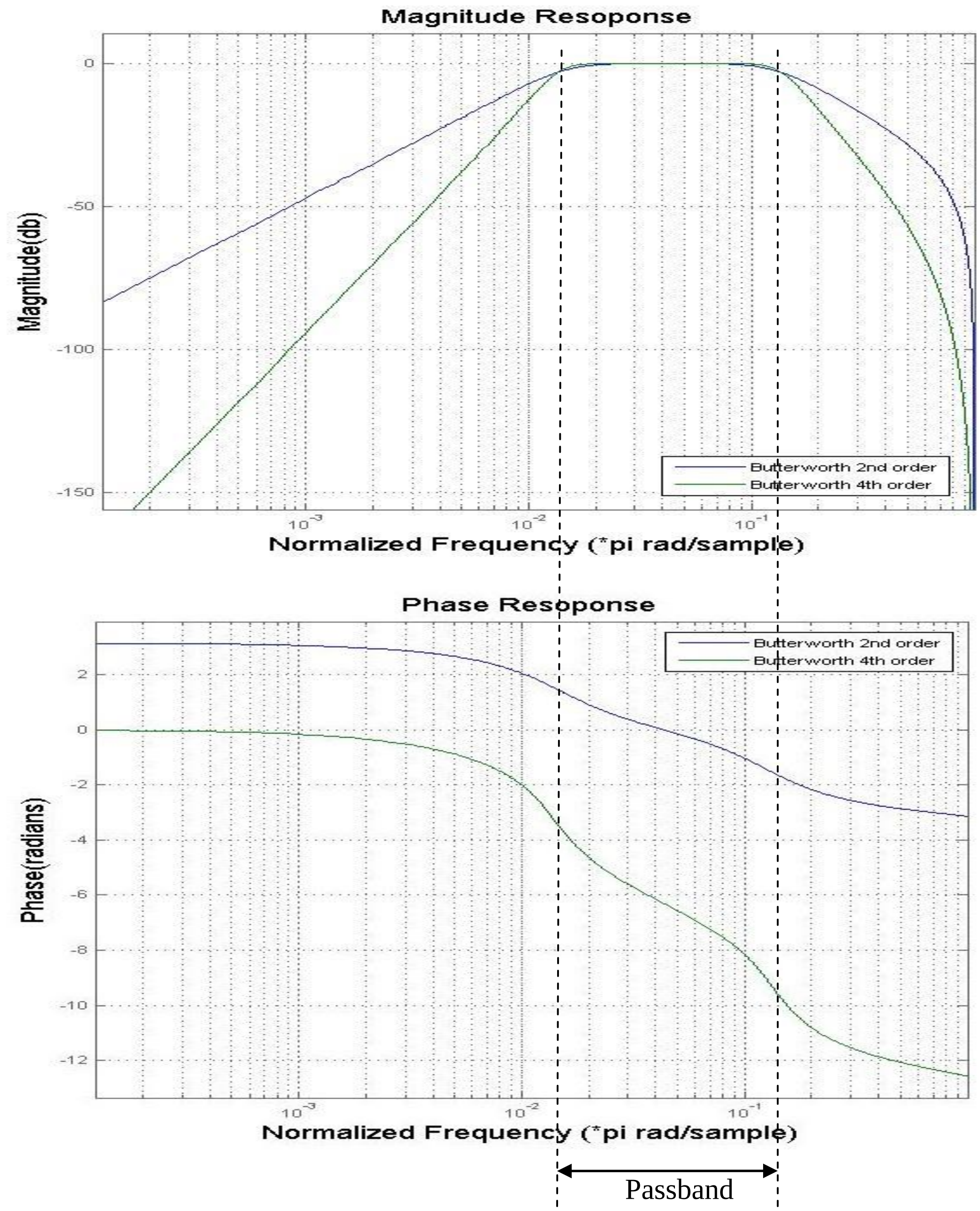


Figure 2-6: Magnitude and Phase response of second order Butterworth and fourth order Butterworth Filter

3 Speech Segmentation

Speech Segmentation is the process of dividing audio signals into voice and unvoiced portions by determining portions of speech which contain weak or unvoiced sounds, including silence segments. This makes it easier to process the voice and calculate feature characteristics (such as pitch) without being affected by noise that may have been left behind after filtering. A variety of methods have been devised to address the problem of speech segmentation, such as magnitude and Zero-crossing rate (ZCR) [9] &phoneme recognition in Putonghua syllables [10]. The approach in [9] detects if the segment is voiced or unvoiced by segmenting the audio based upon a Zero-crossing rate threshold. This threshold is calculated during the initial voice training to determine the average zero crossing rate for the particular individual. Section 3.2 Ref [10] describes a process used to identify voiced portions by the abrupt change in the time domain between different syllables. This has proved to be successful for Chinese syllables as all Chinese syllables are monosyllabic and each syllable can have at most one unvoiced phoneme. This is not applicable to English language as various syllables contain more than one phoneme and are at times polysyllabic. Also methods involving the use of probability density functions (PDF) and linear pattern classifiers (LPC) have been applied [11]. It proposes that PDFs and LPCs work better than zero crossing rate (ZCR) because this method requires only a few samples to calculate a threshold based on probability when compared to ZCR and Short Time Energy (STE) which require a number of samples during the training phase to find a suitable threshold. Ref [12] presents a very robust method of classifying speech and non-speech. This algorithm relies on energy and spectral characteristics of the signal such as Cepstral features. Ref [12] suggests this process is better than traditional energy calculations by creating a 3 level threshold when compared to the 1 level threshold of traditional energy calculations.

The importance of speech segmentation (voiced-unvoiced) as mentioned previously is due to the fact that this process helps to discard the segments that do not contain productive data (Silence, noise, etc.). Bearing in mind that the process incorporated here is speech segmentation and not word segmentation, the task here is to separate voice segments, containing one or more words without a pause in conversation, from unvoiced segments. The solution to this problem is to run each group of audio data from noise filtration through a phonetic classifier that recognizes the phonemes and inflections in spoken audio. Classifying these based on finite state grammar which helps produce words based on their phonemes, the algorithms implemented in previous research thus separates the audio into their individual words which is a traditional requirement for Natural Language Processing (NLP). But word segmentation is a much harder process to implement than speech segmentation due to the large scale of grammar knowledge required as well as computation requirements to process the audio and run each word through a phoneme classifier. Here, speech segmentation is incorporated in a manner that allows for NLP and is a valuable step in the collective effort of speaker recognition and NLP.

For the purpose of this thesis, voiced/unvoiced speech segmentation is accomplished by a combination of two widely known methods, Short Time Energy and Zero-crossing rate. This process involves first dividing the speech into small groups of 20 milliseconds each using Hamming windows, to mirror the short time spans used by common words in the English language [2].This frame is then processed to find the Short time Energy contained in the window and the Zero-crossing rate, which are then compared to a threshold on both parameters to determine if the portion is voiced or unvoiced. These respective algorithms are further defined in the sections below. Consecutive Hamming windows are implemented with a 5ms overlap to account for the possibility that the window could fall over the beginning or end of a word. This provides for a smoother transition between words.

3.1. Short Time Energy & Zero-crossing rate with Data & Results

[13]Short Time Energy is calculated by summing the square of the audio signal within the current Hamming window. Zero-crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative magnitude or negative to positive.

$$\text{STE} = \sum_{1}^{n}(a(i)^\wedge 2)$$

Where, a=amplitude

n = no. of samples over entire window.

To judge the performance and compare the results of Speech Segmentation using Short Time Energy and Zero-crossing rate in the segmentation process, the intelligibility of the audio sample at the end of processing and the separation of each group of words (before silence/after silence) is required to contain the required speech data with very minor variation from the original speech intended, without unwanted attenuation or loss in speech data. A Short Time Energy calculation algorithm was chosen that first windows the filtered audio sample using Hamming windows mentioned in section 3 introduction. Once the segments were windowed, the power of the signal in each window was squared and summed over the interval of the window. Once the short time energy is calculated, the threshold is pre-chosen over multiple tests/observations to be 0.4. A window with energy less than 0.4 is considered to be a window with only noise. This window was for tests run on the same audio recording setup. It may not be the same for a different audio setup. The results for this algorithm are shown in figure 3.1.

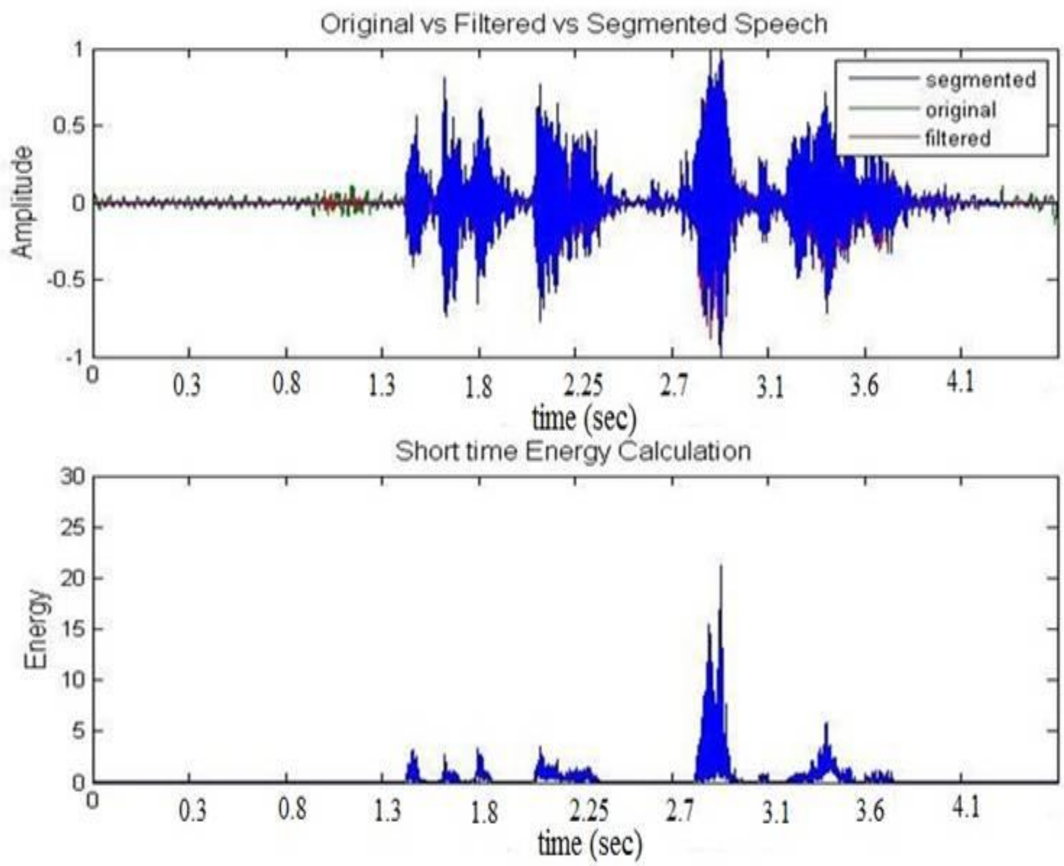


Figure 3-1: Original Speech vs. Segmented Speech

Figure 3-1 shows 4.5 second an audio sample, collected at a sampling rate at 44100. This presents over 200,000samples. The windowing function windows this recording into lengths of 1024 samples (23ms) per window with an overlap of 220 samples (5ms). As we can see, from the above figure, works by quantifying the energy at a given point in time and then assuming voiced segments to have significantly more energy than unvoiced. Also, the energy values above show that, in the sections with only noise, the energy values are at near-zero, in large part due to the filtration module removing unwanted frequencies. While, the segments containing the noise are removed, Figure 3-1 shows each word is intact and these results were consistent with the observed samples from all of the study participants. The error rate for the word is show below.

In the sample audio show in Figure 3-1:

Table 3-1: Data of audio sample used for segmentation test

Data	**Value**
No. of words in current Audio sample	104
No. of seconds of current Audio sample	43.7835 sec
Sampling Rate	44100 samples/sec
No. of samples over entire Audio	1930753 samples
No. of samples in each window	1024 samples
No. of samples of lost over entire audio after segmentation	11000 samples

Table 3-2: STE segmentation results

Words Present for few words in Speech (After filtration)	Percentage of words present after segmentation	Samples of audio present before/after word respectively
Our	100	20/0
Deepest	100	0/0
Fear	100	0/10.88
Is	100	20/0
Not	100	0/0
That	100	0/0
We	100	0/0
Are	100	0/0
Inadequate	100	0/8.27

Note: The 0/0 samples present before and after the word indicates that the word was spoken in continuous speech and the voiced portion was part of the entire segment separated.

In respect to Zero crossing rate, the audio was also windowed using the same Hamming window. Then a counter was initiated within the window that increased every time the signal crossed zero. The average of all counters over the all the speech segments is tagged as the Zero-crossing rate of the individual. The application to the process of segmentation is that since noise has a lower ZCR than human speech, that average ZCR count is found in each window. A threshold was pre-chosen as 26 by means of observation. Once the threshold is chosen, any segment with the ZCR below the threshold is considered as noise and is discarded leaving the valuable information. In case the audio speech just starts in the segment, although the segment is discarded completely, the audio signal isn't lost because the Hamming window is a sliding window that overlaps. The Hamming window was chosen with a window length of 1024and an overlap of 220. Thus the audio signal remains secure but the noise and silence is removed. The results as shown below were inaccurate with the error rate for multiple words shown below.

For the same sample presented in the energy segmentation process, the percentage of segmentation is given below.

Table 3-3: ZCR segmentation results

Words Present in Speech (After filtration)	**Percentage of words present after segmentation**	**Milli Seconds of audio present before/after word respectively**
Our	65	23.21/0
Deepest	70	0/0
Fear	100	0/46.42
Is	100	23.21/0
Not	100	0/0
That	100	0/0
We	100	0/0
Are	100	0/0
Inadequate	80	0/23.21

Note: The 0/0 samples present before and after the word indicates that the word was spoken in continuous speech and the voiced portion was part of the entire segment separated.

Note: Although ZCR was not utilized to accomplish Speech segmentation, this feature is used in the recognition phase as it is a voice biometric and signifies the vibrations *present in the voice of an individual. STE was not utilized as a feature in the recognition phase as it is a representation of the energy present in a particular segment of audio and in no way a representation of the voice biometrics.*

4. Speaker Recognition

Speaker Recognition can be defined as the identification of the person speaking using the characteristics of their voice (Voice biometrics). Speaker Recognition is one the core parts of this project and deals recognizing an individual for which the system has trained, among multiple trained users and unknown speakers. The output from the previous models of filtration and segmentation culminate in this module to extract the features unique to an individual and which help the system to recognize an individual based on his voice biometrics. This system does not interact with the Natural Language Processing (NLP) module but solely deals with recognizing the speaker. The approach studied in this section, considers that the system has certain time to interact with the user (training) so as to be able to associate the speaker with his voice biometrics. It incorporates a feed forward Artificial Neural Network that utilized the initial training speech voice biometrics to create training pairs and target pairs that form the structure of the neurons in the feed forward network as described in detail in section 4.2. These neurons undergo a firing process based on probability and classify the live speaker under the most probable tag.

Ref. [14] is a detailed paper that discusses and compares the various attempts at Speaker recognition. It lists the various methods in which this problem was attempted. The most common methods used, involve implementing machine learning algorithms and stochastic methods that allow the machine to adapt to various feature vectors possible over multiple personalities of voice. One such method is Hidden Markov models. This was employed to allow the computer to train for unforeseen traits that occur due to emotions or urgency that cause a sudden change in pitch. Other such methods employed are wavelet based transforms [16], Artificial Neural networks [15] and support vector machines. These papers proved that HMM and neural networks have similar results with slight differences and similar error rates. Ref. [14] states that in recent years, Support vector machines were being adopted in comparison to Artificial Neural Networks, although they both achieve comparable results (SVM sometimes proving to have better results). Thus, Ref. [17] & [18] employed both Hybrid Hidden Markov models and Neural Networks together. This proved to have low error rates but slower computational speed when compared to each algorithm individually.

For the sake of this thesis, Artificial Neural Networks have been employed to solve the problem of Speaker Recognition. The reason ANNs were chosen was due to the lesser computation speed required when compared to HMM although HMM are better able to compensate for possible hidden states that the ANN. Although, SVMs and HMMs are comparably easy to implement with its own drawbacks, ANN was chosen to prove its functionality so as to apply the usage of neural networks in the NLP model in series with an HMM model. Section 5 further explains HMM models in details including advantages, disadvantages, requirements in terms of data and possible outcomes.

The basic process incorporated in this project for speaker identification is as listed below. It involves collecting filtered and segmented audio data from the previous modules, upon which feature vectors are extracted. This is then used to train the Artificial Neural Network (ANN). Once the ANN is trained, it should distinguish the user from a group of unknown participants. The steps involved in this process are listed below and further descriptions are noted in the sections following.

Steps:

a. Audio Recording for Training
b. Noise reduction, speech segmentation and Feature extraction from recorded audio
c. Neural network, training group and target group assignment
d. Listen to audio from user.
e. Run neural network on new audio with respect to trained set.
f. Collect output.

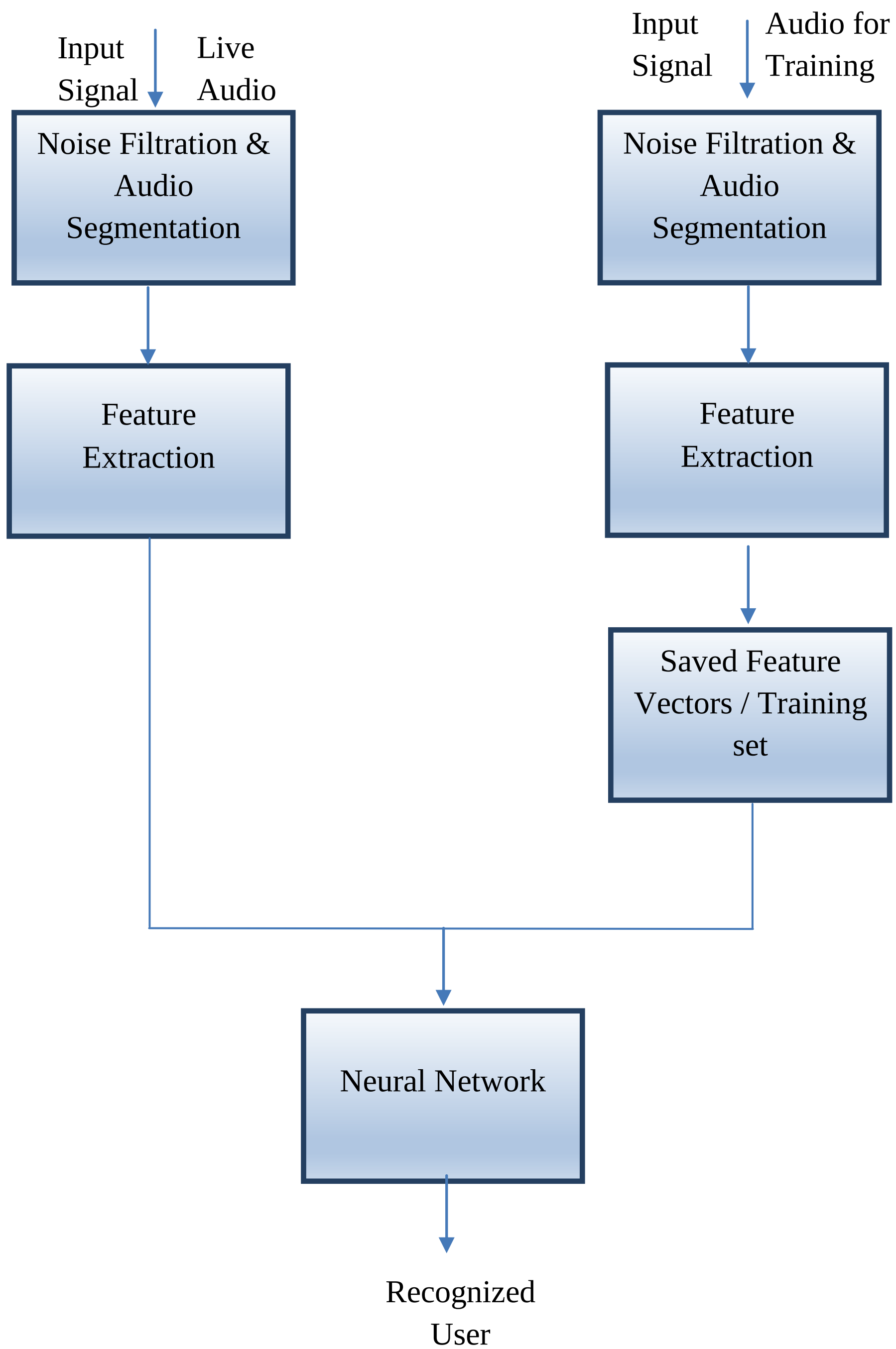


Figure 4-1: System architecture for Neural network

4.1. Data Collection Process

A set of 7 voluntary participants were chosen: 5 male and 2 female. A formal request with the Internal Review Board (IRB) was posted to allow the use of an individual's audio pattern purely for the purpose of training and research guaranteeing the safety and security of the recorded samples. This request is noted in Appendix A. The samples would then be destroyed a year after completion granting the safety of the samples. These samples were stored on board secured drives and were only tagged with the sex and age of the individual.

The participants were seated in a conference room where they were asked to read a paragraph of 100 words. The length of this paragraph was chosen such that there would be a minimum of 70 words for training and 4 sentences for testing for each participant. Also, the participant was requested to read the paragraph three times, to get rid of issues cause by unforeseen noises or sudden change in pitch. The paragraph listed in Section 8.1, Appendix A was chosen for two reasons,

1) The diversity in the English statements used, involving questions and statements that portray belief and confidence. This is important due to the inflections and change in tone caused due to the way statements and questions are spoken. It helps cover portions of voice that may hold different pitch due to the above mentioned inflection.
2) Due to the familiarity of the text and the popularity of the speech.

Feature Extraction is defined as the process of determining certain characteristics of the signal. Every voice has different characteristics that define it, such as Pitch, Zero-crossing rate, Amplitude, Rate of Speech, etc. In this project, two methods for calculating pitch were considered: Auto-correlation and Cross-correlation.

Auto Correlation measures the correlation of a signal with itself shifted in phase with a time lag T. It is represented as

$$f(t) : f(t+T)$$

where, f(t) is the original signal

f(t+T) is the signal with lag T.

Cross-Correlation measures the degree of correlation by correlating the input signal with signals of varying frequency and phase. It is represented as

$$f(\omega_{os}t+\Theta i) : g(\omega_{vs}t+\emptyset)$$

The audio segments from the speech segmentation block are then used to calculate the Pitch.

The Zero-Crossing rate was also calculated by the algorithm previously discussed in Section 3.1. Once these features were extracted, they were tabulated as illustrated in Table 4.1-1 and used in training for an Artificial Neural Network which will be explained in section 4.2.

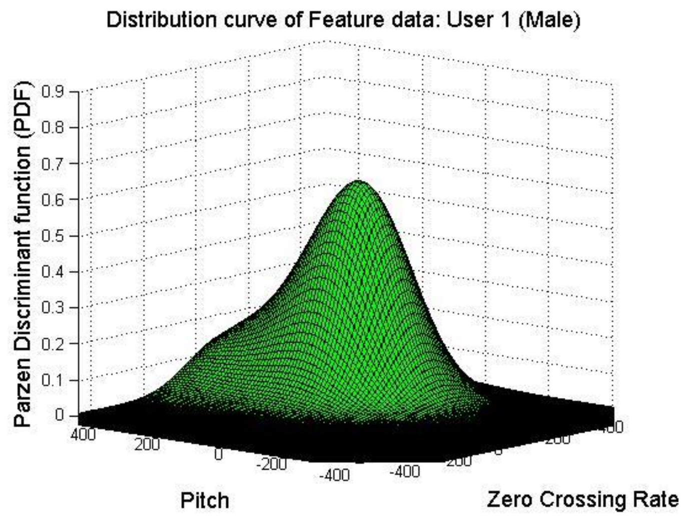
In this project, the Pitch and the Zero-crossing rate (ZCR) were selected for feature extraction. Amplitude calculations were considered inconsistent due to sudden changes of sound levels caused by sentiment, ambient noise etc. Rate of speech calculation were deemed inconsistent as well due to changes caused by sense of urgency or importance, sentiment, etc. Whereas, in Pitch and Zero-crossing rate mentioned in section 3.1, due to the high margins each sample showed between samples of other individuals and, various recordings of each individual showing similar readings, these methods were chosen. The data for the audio recorded are as shown below.

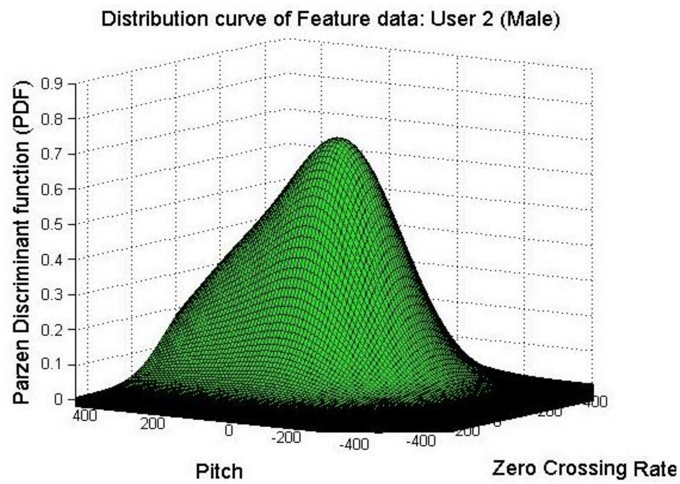
Table 4-1: Table showing ZCR and Pitch data with multiple calculation methods

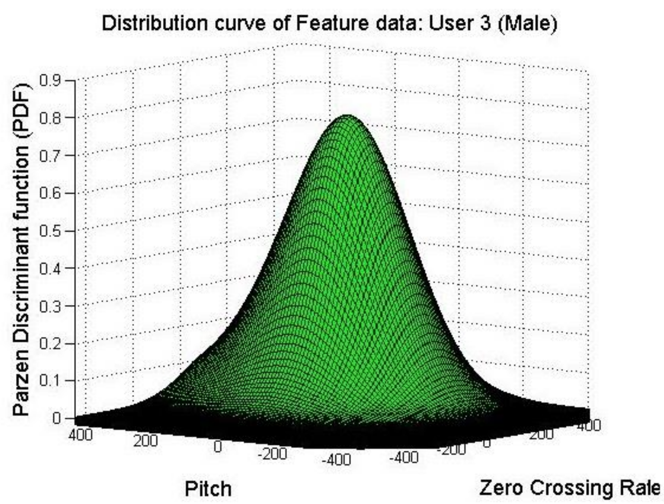
Individual	ZCR	ZCR with 7 point averaging	Pitch using Auto-Correlation (Hz)	Pitch using Cross-Correlation (Hz)
Male 1 (1st try)	196.067	166.056	329.104	687.579
Male 1 (2nd try)	221.032	187.066	134.043	869.819
Male 2 (1st try)	166.952	139.314	352.8	465.546
Male 2 (2nd try)	200.026	164.786	122.161	658.864
Male 3 (1st try)	237.365	199.208	290.132	1100.329
Male 3 (2nd try)	222.368	188.235	112.5	1067.24
Male 4 (1st try)	273.313	230.557	501.136	1193.17
Male 4 (2nd try)	237.514	201.121	100.91=5	1245.93
Male 5 (1st try)	324.366	260.97	128.198	717.1
Male 5 (2nd try)	202.866	168.853	134.043	456.182
Female 1 (1st try)	276.988	239.649	233.333	917.664
Female 1 (2nd try)	266.758	233.801	227.32	944.667
Female 2 (1st try)	187.038	166.256	380.172	571.409
Female 2 (2nd try)	215.271	191.158	205.116	825.905

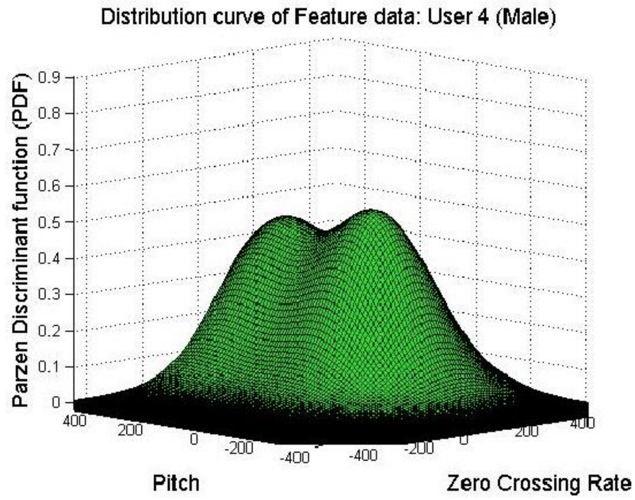
As you can see in the above table, the data for normal ZCR and Pitch using Cross-Correlation seem to be better advocates of a person's characteristics than ZCR with 7-point averaging and Pitch using Auto-Correlation. The data for an individual over multiple tries shows that it follows a possible probability distribution in the pitch and with very minor changes in ZCR. Over multiple individuals, the data fluctuates as they speak but remain within a distinct range, so as to distinguish between these individuals. This differentiation is leveraged in the neural network mentioned in section 4.2 to classify between individuals.

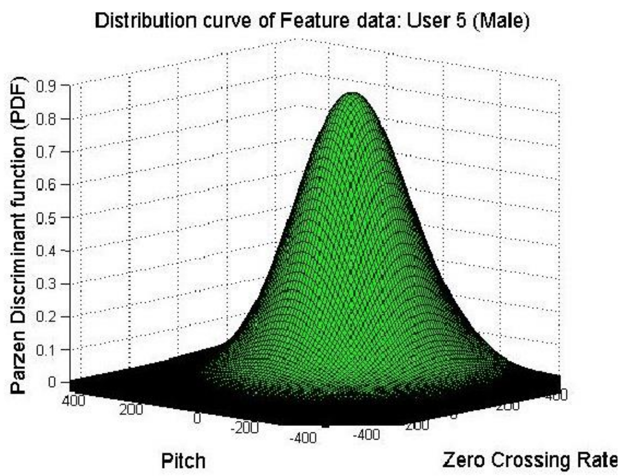
Pitch is defined as a perceptual property that allows the ordering of sounds on a frequency-related scale.[19]For Human speech, the usable voice frequency bands ranges in telephony range from approximately 300 Hz to 3000 Hz.[20][21]There are a variety of pitch detection algorithms that are designed to calculate the Pitch of a speech signal. The Yin algorithm and maximization of the posterior marginals (MPM) algorithm are both based upon autocorrelation in the time domain. Frequency Domain approach involves utilizing the periodogram to convert the signal to an estimate of the frequency spectrum.

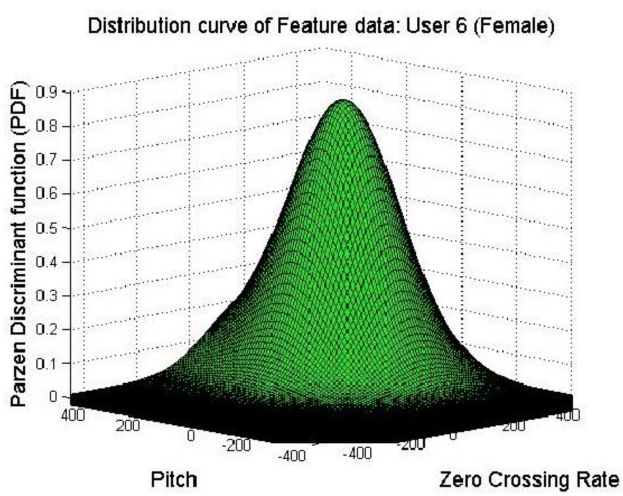

Distribution curve of Feature data: User 1 (Male)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate


Distribution curve of Feature data: User 2 (Male)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate


Distribution curve of Feature data: User 3 (Male)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate


Distribution curve of Feature data: User 4 (Male)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate


Distribution curve of Feature data: User 5 (Male)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate


Distribution curve of Feature data: User 6 (Female)
Parzen Discriminant function (PDF)
Pitch
Zero Crossing Rate

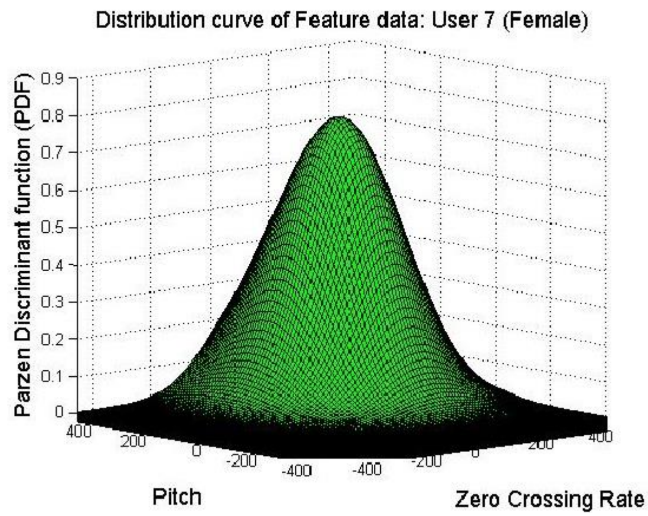


Figure 4-2: Parzen distribution curve for the Zero Crossing Rate & Pitch for each user

As seen in figure 4-2, the probability distribution shows the distribution for each user involving the Zero-crossing rate (ZCR) and Pitch. From this figure it can be seen that the most likely combination of ZCR and pitch is distinct for each user. This is seen in how the peak of each distribution is located at a unique combination of ZCR and pitch. It can also be seen that these distributions form appropriate decision boundaries between the classes by selecting the class with the largest discriminant value for a given ZCR and pitch combination. These figures show that the ZCR and pitch have distinct distribution to distinguish them from each other.

4.2. Neural Networks

Artificial Neural networks are models that are designed to replicate the nervous system of living beings. They generally consist of interconnected neurons which can compute values from input data. The advantage to Neural Networks is their ability to learn and adapt to new information, thus helping with pattern recognition and classification. Other advantages of Neural Networks include Self-Organization, Real-time operation, and Fault Tolerance. [23]The Neuron within a neural network can be defined as a tiny system with multiple inputs and a single output. The Neuron consists of two modes of operation: training mode and Comparison mode. During the training mode, it is trained to associate outputs to inputs. These neurons can be evolved by placing weights on the inputs thus giving it a higher learning curve with respect to those inputs. This method is called the McCulloch and Pitts model (MCP).

Neural Networks can be of many types namely, Feed-Forward Neural Networks, Recurrent Neural Networks, Modular Neural networks, etc. The Neural network used for this project was a Radial basis Probabilistic Neural Network derived from Radial Basis Function Networks and the feed Forward Probabilistic Neural Network. This possesses the characteristics of both networks.

Figure 4-1 below shows the architecture of a probabilistic neural network. The first layer as explained below is the layer that contains the multiplier for the outputs from the Radial Bias function and the Input weights sub-layer. The second layer is one that contains the target weights. This is sent to a function that classifies the input vector to a specific class that has the maximum probability as below

During the training phase, a set of training vectors are extracted from the recorded audio samples mentioned in section 4.1, that define the characteristics of the target. When a live input is fed into the system, the first layer of neutrons computes the distance from input vector to the training vectors created during the training phase. A vector that indicates how close the input is to the training input is then created. The second layer sums the contribution for each class of inputs and thus produces a vector of probabilities. Ultimately, a competitive function picks the maximum of the probabilities and identifies the input vector by tagging a 1 for that class and 0 for other classes.

As shown in Figure 4-1, [8] it is assumed that there are Q input vector/target vector pairs. Each target vector has K elements. One of these elements is 1 and the rest are 0. Thus, each input vector is associated with one of K classes.

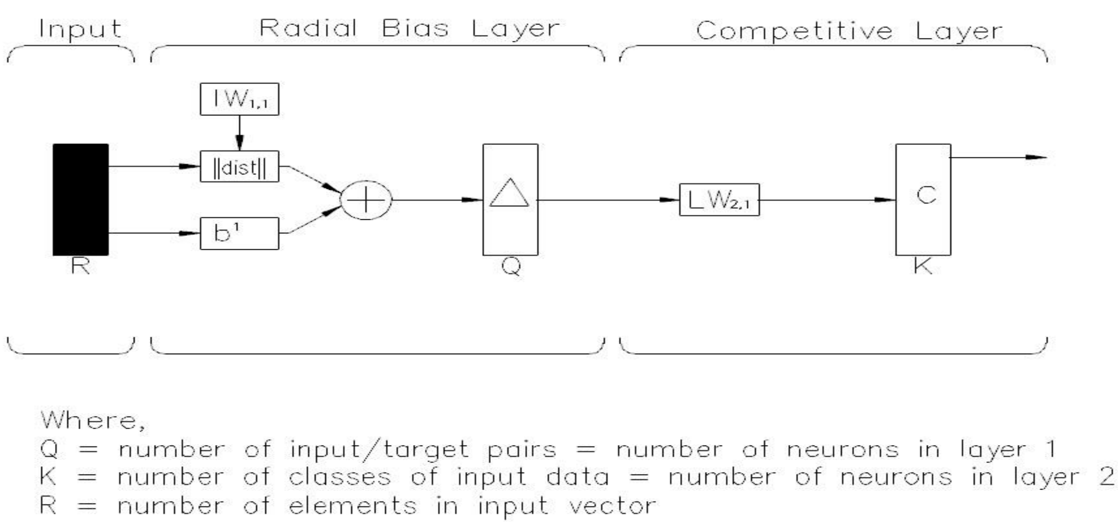


Figure 4-3: Neural Network Architecture
Source: mathworks.com/help/nnet/ug/probabilistic-neural-networks.html

The first-layer input weights, $IW^{1,1}$ (net.IW{1,1}), are set to the transpose of the matrix formed from the Q training pairs, P'. When an input is presented, the ‖ dist ‖ box produces a vector whose elements indicate how close the input is to the vectors of the training set. These elements are multiplied, element by element, by the bias and sent to the Radial Bias function. An input vector close to a training vector is represented by a number close to 1 in the output vector a1. If an input is close to several training vectors of a single class, it is represented by several elements of a1 that are close to 1.

The second-layer weights, $LW^{1,2}$ (net.LW{2,1}), are set to the matrix T of target vectors. Each vector has a 1 only in the row associated with that particular class of input, and 0s elsewhere. The multiplication Ta1 sums the elements of a1 due to each of the K input classes. Finally, the second-layer transfer function, compete, produces a 1 corresponding to the largest element of n2, and 0s elsewhere. Thus, the network classifies the input vector into a specific K class because that class has the maximum probability of being correct.

4.3 Data & Results:

During the Implementation of this module, the ZCR and Pitch were used as the inputs for the training sequence and to design the Probabilistic neural network. Data is presented below.

No. of Individuals: *7 (5 male & 2 female)*

No. of Feature Inputs: *2 (ZCR & Pitch)*

No. of Outputs: *7Binary outputs stating '1' for recognized user and '0' for other users.*

Window length for feature extraction: *88200 samples per window.*

(This was chosen so as to get a better ZCR and pitch calculation across the entire segment and to account for unexpected changes. Since this feature isn't being used currently for segmentation, and only for Network training, the short window length of 101 samples as in section 3.1 was utilized.)

Sampling Rate: *44100 samples per second*

Total segments in each audio: *32-24 segments. (No. of segments differed due to the varying rate of speech of each individual.)*

No. of total training samples for neural network over 7 individuals: *187*

Figure 4-3 shows the neural network implemented in MATLAB.

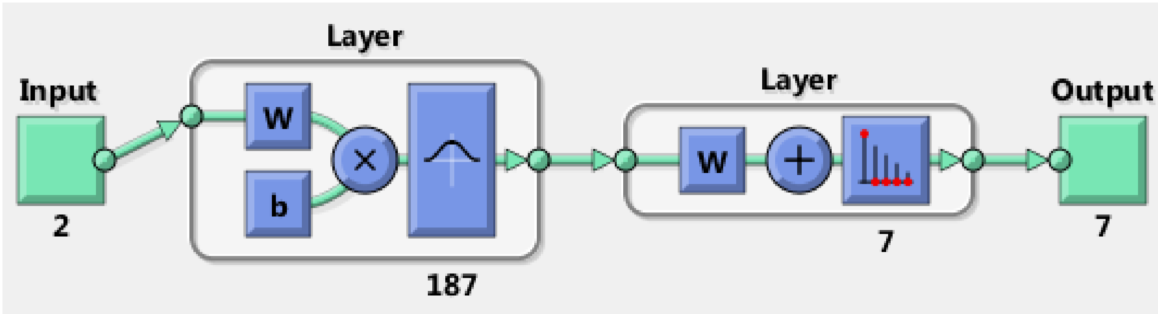


Figure 4-4: Probabilistic Neural Network for Speaker Recognition

(Implemented in MATLAB)

With above data utilized for training sample, the neural network used the calculated pitch and ZCR as the target pairs and the neural network utilizes the radial bias probability to estimate the most probable speaker tag.

Table 4-2: Confusion Matrix (Error Matrix) for Neural Network

		Predicted Class						
		Male					*Female*	
		User 1	*User 2*	*User 3*	*User 4*	*User 5*	*User 6*	*User 7*
Actual Class	*User 1 (Male)*	5	0	2	0	0	0	0
	User 2 (Male)	0	7	0	0	0	0	0
	User 3 (Male)	0	0	6	1	0	0	0
	User 4 (Male)	0	0	1	6	0	0	0
	User 5 (Male)	1	0	2	0	4	0	0
	User 6 (Female)	0	0	0	0	0	6	1
	User 7 (Female)	0	0	0	1	0	0	6

These results are indicative of momentary recognition, where the system recognizes the user after an audio sample of 5 seconds is required and recorded. This is immediately run through all modules to classify the user. This type of recognition resulted in a recognition rate of 85 percent in female audio and 80 percent in male audio. To improve accuracy, another method was implemented, where the audio data was recorded and simultaneously run through all modules to return the classification of User every second. As every second passes, the systems utilized the classification output of the neural network and averages it with previous 4 results and the current neural network result to return a higher accuracy of classification. This method was termed continuous recognition. This was done since it was assumed that the speaker would not abruptly change within a span of 5 seconds. This method resulted in an accuracy of 90 percent in terms of both male and female audio. It resulted in over a sample of 30 seconds, the accuracy for each user reached 90 percent after the first 8 seconds, i.e., after the first 8 seconds, 22 out of 25 results provided the correct user tag. As seen in Figure 4-5, the test show that the values for a recognized speaker converge at the 8th second sample as seen below due to the averaging function explained above.

Note: Incase, the system was interrupted by another user, the system would start returning the result with 90 percent accuracy after 8 seconds, since 4 previous samples and the current sample were required to override the averaging function.

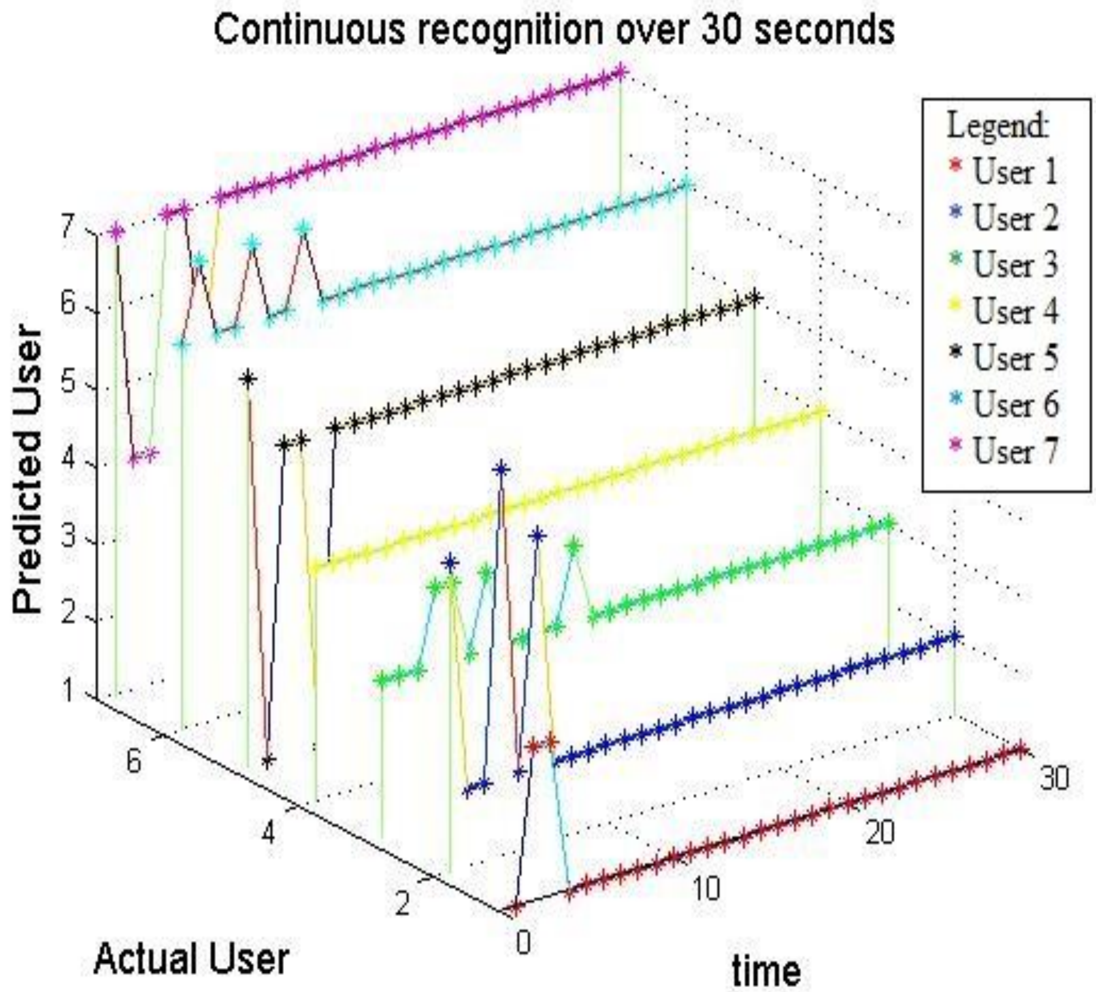


Figure 4-5: Continuous Recognition for Each user over 30 seconds.

Note: In case of an unknown user that the system hasn't trained for, the system would output the closest match to the trained users. This is not a valid option for uses in security application. Hence a probability threshold could be implemented that would classify the user as unknown. This was not implemented in this thesis due to classification requirements faced with large data for multiple user, which would be needed to determine an appropriate threshold.

Note:- The system above is assumed to be language independent as the characteristics used here are Pitch and Zero-crossing rate which remain constant to an individual. This would not be valid if the system utilized the rate of speech or phoneme pronunciation characteristics as its classifiers as these characteristics differ over every language.

5. Language Modelling

NLP is the final module in this thesis and deals with processing the filtered and segmented audio from the first 2 modules. This module has not been implemented due to computational requirements and large database requirements. The method mentioned is only a proposed method and the effects mentioned thereafter are speculated. The data is run through a speech to text converter and the translated text is then processed and tagged with the parts of speech. This section deals with proposing a method of utilizing Artificial Neural networks and Hidden Markov models in series configuration with the output of the neural networks used to assign the transition probabilities and the initial state in the Hidden Markov model.

Existing research has divided the process of NLP into multiple tasks, a few of which are listed below in Table 5-1. These tasks are a general basis for NLP tasks and should not be considered as a final list for the tasks that define NLP.

Table 5-1: Few Natural Language Processing tasks [25]

Task	Description
Automatic summarization	Produce a readable summary of a chunk text
Conference Resolution	Determine which words refer to the same objects
Machine Translation	Automatically translate text from one human language to another
Named Entity Recognition	Determine which items in a text map to proper names and the type of each name (e.g. Person, Location, Organization etc.)
Natural language Generation	Convert information from computer databases into readable human language.
Part of speech Tagging	Given a sentence, determine the part of speech of each word.
Word tense association	Given a sentence, determine the tense of the sentence by comparing the tense of each word in the sentence.

The research of this thesis is concentrated on the tasks of word tense association and parts of speech tagging. The reason for selecting these two tasks is because these tasks form a basis for various other tasks such as Relationship extraction, Sentiment analysis, Question answering, Information extraction, etc.

To accomplish these tasks, an above average understanding of the English language is required. The reason for this is to get a sense that manually entering all the words into a database and creating rules for every words would be time consuming and difficult. [26]For instance, second edition of the Oxford English Dictionary contains full entries for 171,476 words in current use and 47,156 obsolete words. To this 9,500 derivative words can be included as subentries. Over half of these words are nouns, about a quarter adjectives, and seventh verbs. This doesn’t include the words used in everyday language around the world with different inflections and meaning in various dialects. Also, English is an inflectional language. This means that a number of words can be used as nouns, verbs or adjectives. With every inflection, the relationship of the word to the rest of the sentence changes as well. Thus it is important for us to understand the reason to implement machine learning algorithms that learn from pre-existing databases and examples.

5.1. Relation to previous sections:

To accomplish this task, while simultaneously accomplishing speaker recognition, multiple threading has to be implemented. One thread is utilized to achieve speaker recognition, while the second thread is required for translation of speech-to text and further language processing. Figure 5-1 shows the architecture incorporated for the purpose of this project.

The initial step in this process as seen in the architecture below, involves utilizing the filtered and segmented audio input to text. This involves utilizing a pre-existing speech to text converter with a high confidence rate. One such method on a windows computer is to connect the project implementation software to the windows speech recognition tool via a .net server. The windows speech recognition tool has freely available speech SDK and API named SAPI that can be utilized for this purpose.

Once this task is accomplished the remaining steps involve running the translated text through the algorithms mentioned in section 5.2 and section 5.3 to derive the relational information of the words in a sentence and tag them with their respective tenses and part-of-speech information.

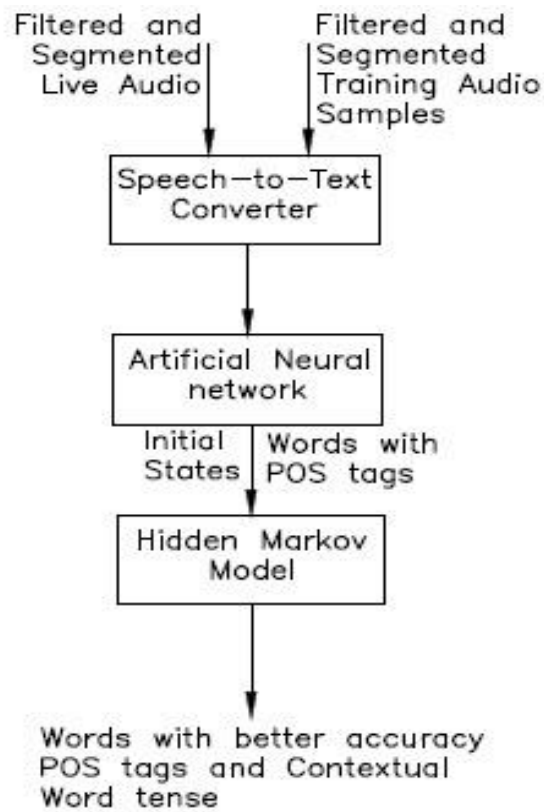


Figure 5-1: NLP module expanded from figure 1-2

5.2. Part-of-Speech Tagging

[30]Part-of-Speech (POS) tagging is defined as the marking of words corresponding to a particular part of speech. This is achieved based on its definition as well as its relationship to words in the same sentence, phrase or paragraph. This is a difficult process, as certain words represent more than one part of speech at different times. In parts of speech tagging using a computer, it is common to distinguish between 50 to 150 separate parts of speech.

Multiple Methods for part of speech tagging have been used. Most tagging algorithms fall into one of two types: rule based taggers and stochastic taggers. Rule based taggers involve a large database of handwritten rules which specify that an ambiguous word is a noun or a verb or some other part of speech. This is performed in two stages. The first stage uses a dictionary to assign each word a list of potential parts of speech. The second stage used large lists of rules to identify the appropriate part of speech. One such database is the Brown Corpus created by Brown University and utilized TAGGIT, the first large rule based tagger that used context-pattern rules. The disadvantage to rule based tagging is the enormous amounts of time taken to create the rules and the processing time is high as it has to run each word through the comprehensive list of rules.

Stochastic based rules resolve tagging issues by using a training database to compute the probability of a given word having a given tag in a given context. Some popular taggers used are Hidden Markov model based, and transformation based taggers

R. Singha, Purkayasta and D. Singha in ref. [31] have incorporated a Rule based POS tagging method that works on the Manipuri Language. They have found that the rule based stochastic method achieves a high accuracy, but it is difficult to classify the lexical categories of Manipuri. This proves that in rule based methodology, it is difficult to assign all the rules of a language to determine the POS tag. Stochastic methods prove to overcome this issue as proved in [32], [33] and [34]. This allows for the incorporation of multiple training algorithms and stochastic methods that greatly reduces the time involved in training as well as incorporates the learning abilities of algorithms such as genetic algorithms and Markov models to assign tags to words not assigned in the rule based method.

Section 5.2 aims to imitate a combination of similar stochastic processes, namely Artificial neural networks (ANN) and Hidden Markov Models (HMM) to tag the given word. The sentence is run through a neural network and a set of words are tagged with a certain accuracy, these words are then run through a HMM to tag the rest of the words by incorporating the Viterbi approximation to gain the most probable state path and give contextual understanding to the statement.

One the input speech is translated and each word is separated, this word is then run through the neural network. As previously explained in Section 4.2 regarding neural networks, this new neural network is initiated by using a training matrix that lists words from an existing finite grammar database (e.g. The Brown Corpus). This training matrix is then utilized to train the neural network. Eighty to eighty five percent accuracy is defined by the fact that most sentences contain words that are always the same part-of-speech in every context. A few examples of these words are verbs such as be, do, etc., and articles such as the, a, an etc.

These words are then added as the input states to form the state space in the HMM, that [35] have been utilized for speaker recognition and Natural Language processing since the mid 1980's. They follow the same structure overall consisting of initial states, transition probabilities, emission probabilities and hidden states. These will be explained in further detail including the architecture of the HMM proposed for the purpose of this thesis.

Hidden Markov models are basically statistical Markov chains that contain hidden (unobserved) processes. In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.

Using the finite state grammar database mentioned earlier, the transition matrix can be generated by the probabilities stated in the database. For example, the Lancaster-Oslo Bergen corpus of British English states that the transition probability from an article to a noun is 0.4, to a verb is 0.4 and to a number is 0.2. The emission matrix can similarly be assigned.

But the outputs from the neural network form an initial state sequence. We can thus generate the transition matrix and the emission matrix given the sequence of states. Once the possible words are assigned by the Artificial Neural Network, the maximum likelihood estimates of transition and emission probabilities are generated. The Viterbi approximation then calculates the most probable state path for the given Hidden Markov model and assigns a tag to these states with accuracy greater than the one achieved by the neural network.

5.3. Word-Tense Tagging

Word tense tagging is the process of tagging a sentence given a preloaded knowledge database. The tense in the sentence gives an understanding of when the situation took place, if it is taking place presently or will take place. [27]Tense is normally indicated by a verb form, either on the main verb or an auxiliary verb. The tense markers are divided into three main categories, and further into sub categories.

The three main categories are past, present and future. To understand these tenses, we need to remember that English is an inflectional language as mentioned in the introduction to section 5. [28]Inflection is the modification of a word to express different grammatical categories. The modification/inflection of a verb is called conjugation. Conjugation is the creation of derived forms of a verb by modifying according to the rules of grammar. These are further divided into subcategories such as past & non-past, immediate future, near future, remote future etc. All these rules are listed in the finite state grammar.

Given the output from the previous section of POS tagging, the contextual sequence of words are given with their POS. Thus leveraging the Parts of speech property, we can use a maximum likelihood algorithm to compare the given part of speech against the finite grammar set to calculate the most probable tense of the sentence.

6. Conclusion & Recommendations:

While re-introducing noise filtration techniques and Segmentation techniques, this thesis incorporates the co-operative effort of all modules together and introduces methods to conduct speaker training as well as Natural Language processing. This process is furthermore implemented in real-time using Matlab. Results indicate that the first four modules, namely filtration, segmentation, feature extraction and speaker recognition work together to give results with an increased accuracy by combining multiple methods. For examples, noise filtration reduces the level of noise to an extent, and with the incorporation of speech segmentation, eliminates noise to a minute negligible level. This in turn increases the accuracy of the pitch determination algorithm and the feature extraction process as a whole. With proper training matrices for the neural network, speech recognition has returned an accuracy of 85percent, implying that 1-2out of 10 tries may fail for momentary recognition and 90 percent accuracy (1/10 may fail) for continuous recognition.

This paper also introduces a method involving the combination of neural networks and Hidden Markov models to tag words with its respective parts of speech and tense, in relation to the text surrounding it. Future work will be focused on implementing this process.

7. References:

1) Speaker recognition. (2014, October 16). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Speaker_recognition

2) Benesty, J., Sondhi, M., & Huang, Y. (2008). Springer handbook of speech processing. Berlin: Springer Verlag.

3) Filter (signal processing). (2014, October 18). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Filter_ (signal_processing)

4) Infinite impulse response. (2014, August 29). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Infinite_impulse_response

5) Sorenson, J., Johannesen, L., Grove, U., Lundhus, K., Couderc, J., & More authors. (2010). A Comparison of IIR and Wavelet Filtering for Noise Reduction of the ECG.Computing in Cardiology, 489-492.Retrieved September 21, 2014, from IEEE Database.

6) McFarlane, A. (1997). Sensor Review. In Digital signal processing (1st ed., Vol. 17, pp. 13-20). MCB UP.

7) Abbas, S. (2011). Comparative studies of different types of an active filter. Journal of Engineering and Development, 15(4), 49-60.

8) Documentation: Probabilistic Neural Networks. (n.d.). Retrieved October 21, 2014, from http://www.mathworks.com/help/nnet/ug/probabilistic-neural-networks.html

9) Daaboul, F., &Adoul, J. (1977). Parametric segmentation of speech into voiced-unvoiced-silence intervals. Acoustics, Speech, and Signal Processing, 2, 327-331.

10) Chan, C. (1986). Voiced/Unvoiced segmentation. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '8i6., 11, 2271-2274.

11) Saha, G., Chakraborthy, S., &Senapati, S. (2005). A New Silence Removal and Endpoint Detection algorithm for speech and speaker recognition applications. Proceedings of the Eleventh National Conference on Communications, 291-295.

12) Bou-Ghazale, S., &Assaleh, K. (2002). A robust endpoint detection of speech for noisy environments with application to automatic speech recognition.2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 4, IV-3811.

13) Zero-crossing rate. (2014, February 8). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Zero-crossing_rate

14) Cutajar, M., Micallef, J., Casha, O., Grechi, I., &Gatt, E. (2013). Comparative study of automatic speech recognition techniques. IET Signal Processing, 25-46.

15) Hennebert, J., Hasler, M., &Dedieu, H. (1994). Neural Networks in Speech Recognition. 6th Microcomputer School, Invited Paper, Prague, Czech Republic, 23-40.

16) Gamulkiewicz, B., & Weeks, M. (2003). Wavelet based speech recognition. 2003 IEEE 46th Midwest Symposium on Circuits and Systems, 2, 678-681.

17) Tang, X. (2009). Hybrid Hidden Markov Model and Artificial Neural Network for Automatic Speech Recognition. 2009. PACCS '09. Pacific-Asia Conference on Circuits, Communications and Systems, 682-685.

18) Trentin, E., & Gori, M. (2004). Robust combination of neural networks and hidden Markov models for speech recognition. IEEE Transactions on Neural Networks, 14(6), 1519-1531.

19) Pitch (music). (2014, October 18). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Pitch_(music)

20) Vocal range. (2014, October 18). Retrieved October 21, 2014, from en.wikipedia.org/wiki/Vocal_range

21) Hayes, M. (1996). Statistical digital signal processing and modeling (1st ed., p. 393). New York: John Wiley & Sons.

22) Michael Noll, A. (1970). Pitch Determination of Human Speech by the Harmonic Product Spectrum, the Harmonic Sum Spectrum and a Maximum Likelihood Estimate. Proceedings of the Symposium on Computer Processing in Communications, 19, 779-797.

23) Stergiou, C., &Siganos, D. (2011, September 24). Neural Networks. Retrieved September 21, 2014, from http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

24) Digital signal processing. (2014, October 27). Retrieved November 12, 2014.

25) Natural language processing. (2014, October 11). Retrieved November 14, 2014, from http://en.wikipedia.org/wiki/Natural_language_processing

26) How many words are there in the English language? (n.d.). Retrieved November 14, 2014, from http://www.oxforddictionaries.com/us/words/how-many-words-are-there-in-the-english-language

27) Grammatical tense. (2014, November 14). Retrieved November 14, 2014, from http://en.wikipedia.org/wiki/Grammatical_tense

28) Grammatical conjugation. (2014, May 11). Retrieved November 14, 2014, from http://en.wikipedia.org/wiki/Grammatical_conjugation

29) List of English irregular verbs. (2014, December 10). Retrieved November 14, 2014, from http://en.wikipedia.org/wiki/List_of_English_irregular_verbs

30) Part-of-speech tagging. (2014, November 14). Retrieved November 15, 2014, from http://en.wikipedia.org/wiki/Part-of-speech_tagging

31) Rajusingha, K., Purkayastha, B., &Singha, K. (2012). Part of Speech Tagging in Manipuri: A Rule based Approach. International Journal of Computer Applications, 31-36.

32) Alba, E., Luque, G., & Araujo, L. (n.d.). Natural language tagging with genetic algorithms.Information Processing Letters, 173-182.

33) Ma, Q., Lu, B., Isahara, H., & Ichikawa, M. (2002). Part of speech tagging with min-max modular neural networks. Systems and Computers in Japan, 30-39.

34) Br, S., & P, R. (2012). Kannada Part-Of-Speech Tagging with Probabilistic Classifiers. International Journal of Computer Applications, 26-30.

35) Murveit, H., & Moore, R. (1990). Integrating Natural Language Constraints into HMM-based Speech Recognition. International Conference on Acoustics, Speech and Signal Processing, 1, 573-576.

36) Zhou, G., & Su, J. (2002). Named entity recognition using an HMM-based chunk tagger. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 473-480.

37) Ma, Q., Lu, B., Isahara, H., & Ichikawa, M. (2002). Part of speech tagging with min-max modular neural networks. Systems and Computers in Japan, 30-39.

38) Schmid, H. (1994). PART-OF-SPEECH TAGGING WITH NEURAL NETWORKS. Proceedings of the 15th Conference on Computational Linguistics, 1, 172-176.

8. Appendix

8.1 Appendix A- Internal Review Board Application & Paragraph used for Audio recitation.

1. Background and Purpose: Briefly describe the background and purpose of the research.

In today's world, with a rapid increase in technology all around us, it has become standard to make certain aspects of every product more open and user-friendly to the end user. With Speech recognition software becoming easily available on the open market, there arises a need for user dependent speech recognition to increase security on specific devices. An accurate example of such a need would be for security systems or plain user recognition on robotic models which are to be trained to the user. The purpose of this research is not only improving on current user recognition methods but also to implement "Natural Language Selection" i.e., training the module to understand the users' command irrespective of the way the user phrases each command. For example, the command "STOP" can be phrased as "Halt", "don't move", and "Abort".

2. Design, Procedures, Materials and Methods: Describe the details of the procedure to be used and the type of data that will be collected.

For the purpose of the user recognition, we make a small library of speech data collected from approximately 10 male (including the Investigator) and 10 female subjects. This is to test the module to distinguish the Main User (Investigator) from the group. The speech data will consist of two paragraphs of approximately 100 words each which will be recorded using a program written in Matlab and stored in wav format on an encoded secure hard drive and password protected on the investigators ERNIE file system as a backup.

3. Measures and Observations: What measures or observations will be taken in the study? If any questionnaires, tests, or other instruments are used, provide a brief description and include a copy for review (computer programs may require demonstration at the request of the IRB).

During the course of the study, speech data consisting of two paragraphs of approx. 100 words each will be collected from each participant. The features extracted from this data will be gender and the actual audio recording. No other data like the name or other identifying characteristics will be kept.

4. Risks and Benefits: Describe any potential risks to the dignity, rights, health or welfare of the human subjects. Assess the potential benefits to be gained by the subjects as well as to society in general as a result of this project. Briefly assess the risk-benefit ratio.

Risk: There is no risk to the health or welfare of the participant. The privacy of the individual is also under no inherent risk. The Benefits: This will result in an improved understanding of voice and natural language selection which will improve efficiency and security of personal electronic devices.

5. Informed Consent: Describe the procedures you will use to obtain informed consent of the subjects and the debrief/feedback that will be provided to participants. See Informed Consent Guidelines for more information on Informed Consent requirements.

Procedure and complete details of the Informed Consent Form are attached with this application.

6. Anonymity: Will participant information be anonymous (not even the researcher can match data with names), confidential (names or any other identifying demographics can be matched, but only members of the research team will have access to that information. Publication of the data will not include any identifying information), or public (names and data will be matched and individuals outside of the research team will have either direct or indirect access. Publication of the data will allow either directly or indirectly, identification of the participants). Justify the classification and describe how privacy will be ensured/protected.

The Participant information will remain anonymous. The privacy of the participants will be ensured as the speech data will not be tagged in any way other than the gender of the participant. Data will be kept on an encoded secure hard drive and password protected on the investigators ERNIE file system as a backup.

7. Privacy: Describe the safeguards (including confidentiality safeguards) you will use to minimize the risks. If video/audio recordings are part of the research, please describe how that data will be stored or destroyed.

Audio recording will stored in an encoded wav format on a secured hard drive and password protected on the investigators ERNIE file system as a backup. It will be destroyed upon completion of the project.

8. Participant Population and Recruitment Procedures: Who will be recruited to be participants and how will they be recruited. Note that participants must be at least 18 years of age to participate. Participants under 18 years of age must have a parent or guardian sign the informed consent document.

Approximately 10 male and 10 female participants among the faculty, staff and students on Embry Riddle shall be recruited based on availability. The age of the participants will be within the range of 18 – 35. Participants will be selected from the Robotics Assoc. of Embry Riddle who are knowledgeable of the study.

9. Economic Considerations: Are participants going to be paid for their participation? If yes, describe your policy for dealing with participants who 1) Show up for research, but refuse informed consent; 2) Start but fail to complete research.

No paid participants.

10. Time: Approximately how much time will be required of each participant?

Approximately 10-15 minutes will be required of each participant to record and verify said recordings.

<u>Paragraph:-</u> *"I have to believe in a world outside my own mind. I have to believe that my actions still have meaning, even if I can't remember them. I have to believe that when my eyes are closed, the world's still there. Do I believe the world's still there? Is it still out there? Yeah. We all need mirrors to remind ourselves who we are. I'm no different." - Memento*

8.2 Appendix B – Speaker Recognition and Audio feature System Graphic User Interface

The figures below show the actual system user interface that was implemented in this thesis.

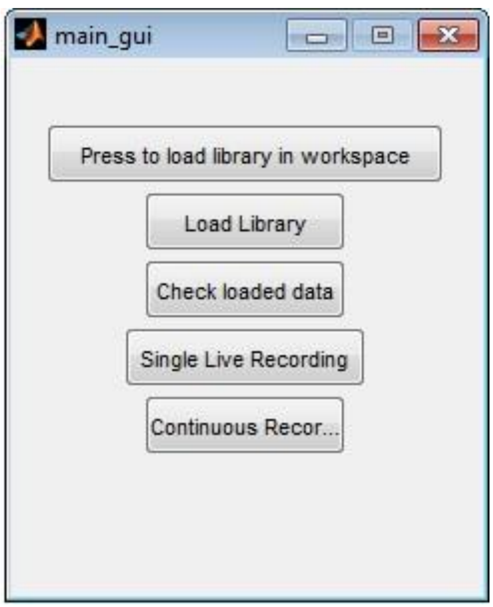


Figure 8-1: Main Screen GUI for SR module

<u>Main GUI (as shown in Figure 8-1):</u>

<u>*Option 1:*</u> To load the library into MATLAB workspace for testing/verification operations outside of GUI thread.

<u>*Option 2:*</u>To load the library of data for each user.

Data: Audio Data (trial 1, trial2), name tag (ex. Male1, Female2, etc.), ZCR (for each segment), Pitch (for each segment).

While processing, the following lines are displayed on the Command window to alert the User of operations underway or completed.

"Fragmentation & calculating required data"

"Calculating Zero Correlation and Pitch"

"Assigning test and target groups for Neural Network"

"Entire Library Loaded"

<u>*Option 3:*</u>To check individual user feature data (shown in Figure 8-2).

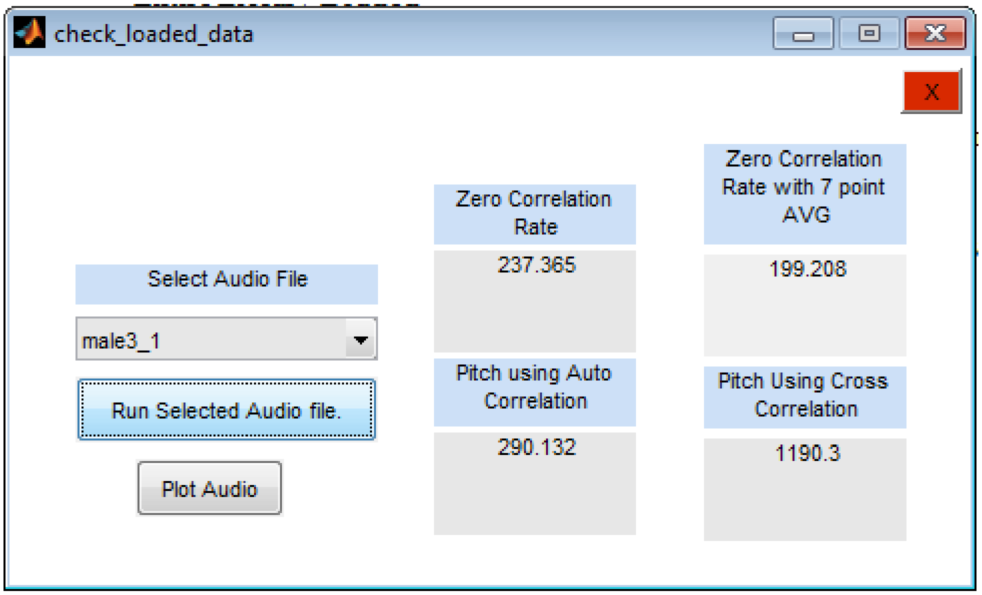


Figure 8-2: GUI to check feature data of individual user

The drop down box allows the selection of a user based on the tag and the plot audio option plots the audio sample for verification and comparison.

Upon pressing the red 'X' button, it returns to the main GUI.

*Option 4:*To check for recognized user tag on a momentary single live recording (shown in Figure 8-3).

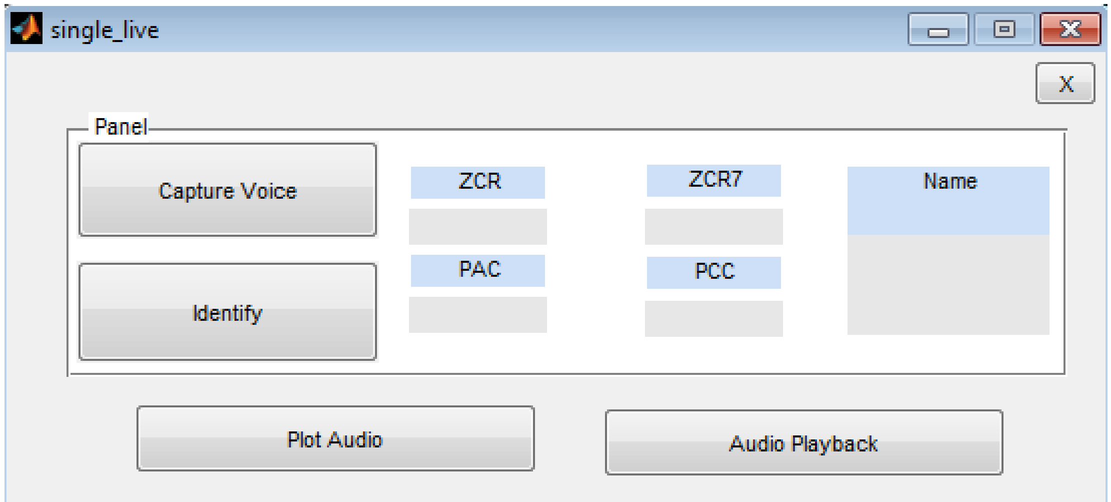


Figure 8-3: GUI to record single live momentary recording and recognize user.

The "Capture Voice" option opens another GUI as shown below to start the recording process and end the recording process.

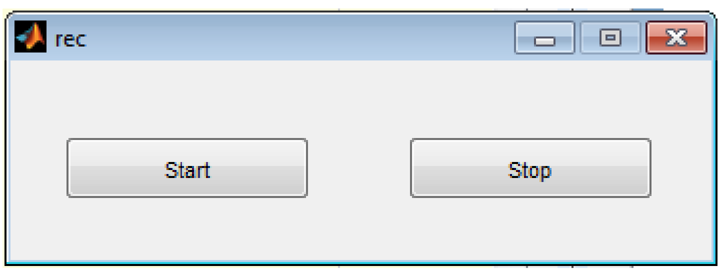


Figure 8-4: GUI to start and stop momentary recording.

Open pressing the Identify button in the "single_live" GUI, the system calculates the various features as shown in Figure 8.2-3 along with the recognized user tag in the "Name section". The other options are to plot the recorded audio and to play the recorded audio to verify the signal.

Upon pressing the red 'X' button, it returns to the main GUI.

*Option 5:*To check for recognized user tag on continuous live speech(shown in Figure 8-5).

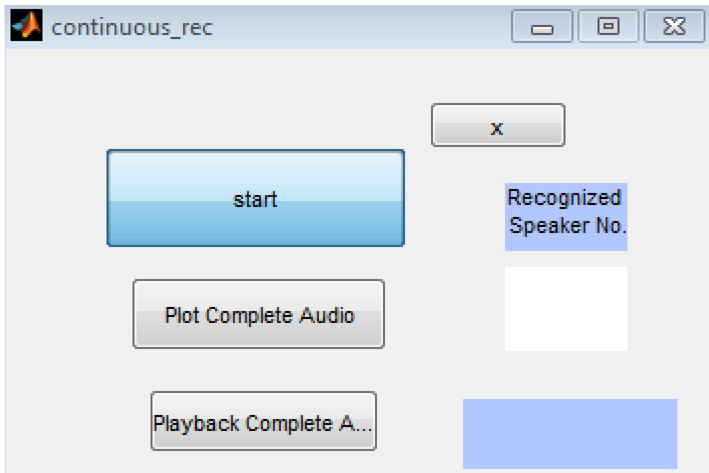


Figure 8-5: GUI for continuous speaker recognition.

The “start” option initiates the system’s microphone device and continuously listens for audio data. It simultaneously (multiple-threading), calculates the audio feature data and updates the user tag in the “Recognized Speaker No.” position every 5 seconds. A time span of 5 seconds was chosen to collect a sentence or two and to give the system sufficient time to average the feature data.

The option to plot and playback the audio are also present for verification purposes.

Upon pressing the red ‘X’ button, it returns to the main GUI.