Sudan University of Sciences and Technology

Faculty of Graduate Studies

Design of a Model for Right Pronunciation of Arabic Letters using Machine Learning Techniques

تصميم نموذج للنطق الصحيح للحروف العربية بإستخدام تقنيات تعلم الآلة

Research Summited in Fulfillment of the Requirements for Master Degree in Information Technology

By

Abeer Mohammed Kheir Osman Ismail

Supervisor: Dr. Mohamed Adany Hamdd Sayed

July 2019

Dedication

I would like to dedicate this thesis mainly to our parents who have helped me and encouraged me to push through the hard times that I have faced throughout my life.

To all those who have stood on my sides and made me believe in myself, I would like to offer a special thanks to them.

Acknowledgements

Firstly, I would like to thank Allah (the Most Gracious the Most Merciful) for everything that have been accomplished, without him nothing could have been done.

The outcome of any project work depends mainly on the education received, the research resources and the excellence of the teachers.

First, my gratitude goes out to my supervisor, Dr. Mohamed Adani who was the cause of my encouragement and success.

Secondly, I would like all the teachers who had an active role in help and provision.

Third, special thanks to my husband who assisted with my thesis writing and helped me with my software integration.

Lastly, I would like to offer a special thanks to all of our colleagues, friends and family for their reinforcements and comprehensive support

Table of Contents

List of Abbreviations

No.	Abbreviation	Meaning
1	A/D	Analogue to Digital Conversion
1	ASR	Automatic Speech Recognizer
2	DCT	Discrete Cosine Transform
3	DFT	Discrete Fourier Transform
4	FFT	Fast Fourier Transform
5	FIR	Finite Impulse Response
6	GUI	Graphical User Interface
7	HMM	Hidden Markov Model
8	HPF	High Pass Filter
9	Hz	Hertz
10	Im	Imaginary
11	Log	Logarithm
12	LPC	Linear Predictive Coding
13	MATLAB	MATRIX LAB
14	MFCC	Mel Frequency Cepstral Coefficients
15	MSE	Mean Square Error
16	NN	Neural Networks
17	PLP	Perceptual Linear Prediction
18	PSD	Power Spectrum Density
19	Re	Real
20	STE	Short Term Energy

List of Tables

List of Figures

Abstract

Automatic speech recognition (ASR) plays an important role in taking technology to the people. There are numerous applications of speech recognition such as direct voice input in aircraft, data entry and speech-to-text processing. The aim of this research was to develop a voice system to learn Arabic letter pronunciation based on machine learning algorithms.

ASR system can be divided into three different phases: signal preprocessing, feature extraction and feature classification. MATLAB platform was used for feature extraction of voice using Mel Frequency Cepstrum Coefficients (MFCC). Matrix of MFCC features was applied to back propagation neural networks for Arabic letter features classification. The overall accuracy obtained from this classification was 65% with an error of 35% for one consonant letter, 87% accuracy and an error of 13% for 10 isolated different letters and 6 vowels each and finally 95% accuracy and an error of 5% for 66 different examples of one letter (vowels, words and sentences) stored in one voice file.

المستخلص

تؤدي أنظمة التعرف على الأصوات الرقمية دورا هاما في تعليم اللغات المنطوقة وتعليم الحروف. هناك الكثير من التطبيقات المتنوعة لأنظمة التعرف على الصوت بإستخدام جهاز الحاسوب مثل الطائرات وأنظمة البيانات وعمليات تحويل النص الى صوت. الهدف من هذا البحث هو تطوير نظام صوتي لتعليم الحروف العربية للأطفال بإستخدام تعلم الآلة. يمكن تقسيم أنظمة التعرف على الصوت الى ثلاث مراحل رئيسية هي: معالجة الإشارة الرقمية واستخراج خصائص الصوت وتصنيف الصوت.

تم استخدام منصة الماتلاب كأداة هامة لإستخراج خصائص الصوت المعتمدة بناءاًعلى تقنية (معاملات سبسترم لترددات ميل). تم تدريب الشبكة العصبية على هذه المعاملات المستخرجة من الاشارة الصوتية الداخلة حيث تم حساب دقة التصنيف النهائي لنطق الحرف لتصل الى 65% وبلغت نسبة الخطأ الكلي للتصنيف 35% لحرف واحد ساكن وبلغت دقة التصنيف النهائي 87% ونسبة خطأ 13% لعشرة أحرف مختلفة بست حركات لكل منها بينما بلغت دقة التصنيف 95% ونسبة خطأ 5% عندما استخدم 66 مثال مختلف لحرف واحد من حركات وكلمات وجمل تم تسجيلهم في ملف صوتي واحد.

CHAPTER ONE: INTRODUCTION

1.1 Overview

Arabic language is a Semitic language, and it is one of the oldest languages in the world. Currently it is the fifth language in terms of the number of speakers [1]. Arabic is the important language especially for Muslims because it is the Qur'an language, so that must be learn very well especial l pronunciation of letter because there are some letters have similar *makhraj* and features. *Makhraj* is the place of the letter out [2].

All Muslims when recite verses of the holy Quran in praying or in other situations, they should follow the rules of pronunciation. Despite this fact, there has been little research on Arabic speech recognition compare to other languages of similar importance (e.g. Spanish or Mandarin) [1].

The enormous advances in the computer technologies in the twentieth century permitted computers to effectively contribute in various fields of human life [3]. One of these is the speech recognition technology. Speech recognition is the process in which the computer will be able to specify the pronouncing word. This means to speak with your computer and then to recognize what you have said properly [4].

The fields of machine learning and pattern recognition provide us with methods to classify data into different groups, fit curves or make predictions. We say learning, because we don't explicitly define a function *f* that gives us a certain output *y*, given an input vector *x*. Problems such as speech recognition and computer vision are too complex to solve analytically. Algorithms can help us to instead solve them iteratively. This means that a large part of learning can be done more or less automatically, but not always entirely automatically [5].

Artificial neural networks is now quite an old subfield of pattern recognition lying on mimicking biological neurons with simple neural networks using electrical circuits [5].

Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain [4]. Two different ways to use neural networks for acoustic modeling, namely prediction and classification of the speech patterns. Prediction is shown to be a weak approach because it lacks discrimination, while classification is shown to be a much stronger approach [4]. Mel-Frequency Cepstral Coefficients (MFCC’s) is a type of algorithm i.e. used to define relationship between Human ear's critical bandwidths with frequency. This method is used for analyzing and extraction of pitch vectors [4].

So, in this project a smart system will be designed to record voice signal, recognize its feature using (MFCC) method and classify it as being the correct the spoken/pronounced letter or not using neural networks.

1.2 Problem Statement

Sudanese people have common and major difficulty with pronouncing some Arabic letters such as: (“ظ” “ض”) and (“ت” “ط”), because they have relative acoustic outputs and features.

In addition, primary students and next generations have the same problem in pronouncing letters wrong, so there is a high needs to design an accurate computer learning system for students of primary levels, Quranic schools (Khalawi) and also non-native speakers to get efficient and clear pronouncing of Arabic letters.

1.3 Research Significance

Learning the pronouncing of some Arabic letters properly will give a higher ability to read Quran correctly without giving wrong meaning for verses, also some Arab people do not know the Sudanese dialects and this system can help them to interact easily with and understand this dialects.

It will be helpful for those who are new in Islam and want to learn how to read and recite Holy Quran properly, or other Arabic non-native speakers want to learn Arabic language.

1.4 Objectives

1.4.1 Aim

To design a voice system to learn Arabic letter pronunciation based on machine learning algorithms.

1.4.2 Sub-objectives

1. To compare between the signal preprocessing of standard letter phenome and spoken one.
2. To measure the performance of classification in terms of accuracy.

1.5 Methodology

Various machine-learning algorithms have been developed and enhanced in speech recognition.

A. **Voice data collection:** The database used consists of records of one human voice of once one letter, 10 different letters with 6 different vowels each and finally one letter with 66 different examples.

B. **Signal pre-processing:** The input phenome signal has to be framed, silenced removed and filtered.

C. **Feature extraction:** MFCC (Mel-Frequency Cepstral Coefficients) feature extraction is a method that is widely used in speech recognition.

D. **Classification:** Artificial Neural Network (ANN) classification method is applied to recognize the final output phenome in form of either Yes “Spoken correctly” or No “Not spoken correctly.

1.6 Research Scope:

1. All Arabic letters.
2. MATLAB.

1.7 Thesis Layout

A brief information about the rest of this thesis is presented here as follow:

- **Chapter 2: Theoretical Background and Literature Studies:** This chapter reviews many of related studies to the field of speech recognition systems of Arabic letters and details theoretical background about this system.
- **Chapter 3: Design of Proposed Voice System:** This chapter shows all the work carried out in the project in software levels.
- **Chapter 4: Results and Discussion:** This chapter shows all the results of each phase of proposed system.
- **Chapter 5: Conclusion and Recommendations:** This chapter shows a conclusion for the results obtained, features and limitations of the different methods of implementation.

CHAPTER TWO: THEROATICAL BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

In this chapter, theoretical overview and many related works have been summarized for speech system of human, generation of speech and the speech recognition system.

2.2 Theoretical Background

2.2.1 Spoken Language Structure

Spoken language used from a speaker to a listener for communication process. “Speech begins with a thought and intent to communicate in the brain, which activates muscular movements to produce speech sounds” [14].

The speech production process begins as a message in the mind of speaker to send to listener, after compose the message the next step is to convert the message into sequence of words. Each word consists of a sequence of phonemes that match to the pronunciation of the words. Each group of words also contains a prosodic pattern that indicates the time of each phoneme, tone of the sentence, and loudness of the sounds [14].

2.2.2 Speech Production Process

“Speech is produced by air-pressure waves emanating from the mouth and the nostrils of a speaker. In most of the world’s languages, the inventory of phonemes, can be split into two basic classes:

- Consonants - articulated in the presence of constrictions in the throat or obstructions in the mouth (tongue, teeth, and lips) as we speak “ء ب ت ث ح ج خ ..... ز و ي”.
- Vowels - articulated without major constrictions and obstructions.” “يِ اَ وُ”[14]

2.2.3 Speech communication pathway

The total components of speech pathway are the lungs, trachea, larynx (organ of voice production), pharyngeal cavity (throat), oral and nasal cavity. "The pharyngeal and oral cavities are typically referred to as the vocal tract, and the nasal cavity as the nasal tract. As illustrated in Figure 2.1, the human speech production apparatus consists of:

Lungs: source of air during speech.

- Vocal cords (larynx): when the vocal folds are held close together and oscillate against one another during a speech sound, the sound is said to be voiced. When the folds are too slack or tense to vibrate periodically, the sound is said to be unvoiced. The place where the vocal folds come together is called the glottis.
- Velum (Soft Palate): operates as a valve, opening to allow passage of air (and thus resonance) through the nasal cavity. Sounds produced with the flap open include m and n.
- Hard palate: a long relatively hard surface at the roof inside the mouth, which, when the tongue is placed against it, enables consonant articulation.
- Tongue: flexible articulator, shaped away from the palate for vowels, placed close to or on the palate or other hard surfaces for consonant articulation.
- Teeth: another place of articulation used to brace the tongue for certain consonants.
- Lips: can be rounded or spread to affect vowel quality, and closed completely to stop the oral air flow in certain consonants (p, b, m) [14].

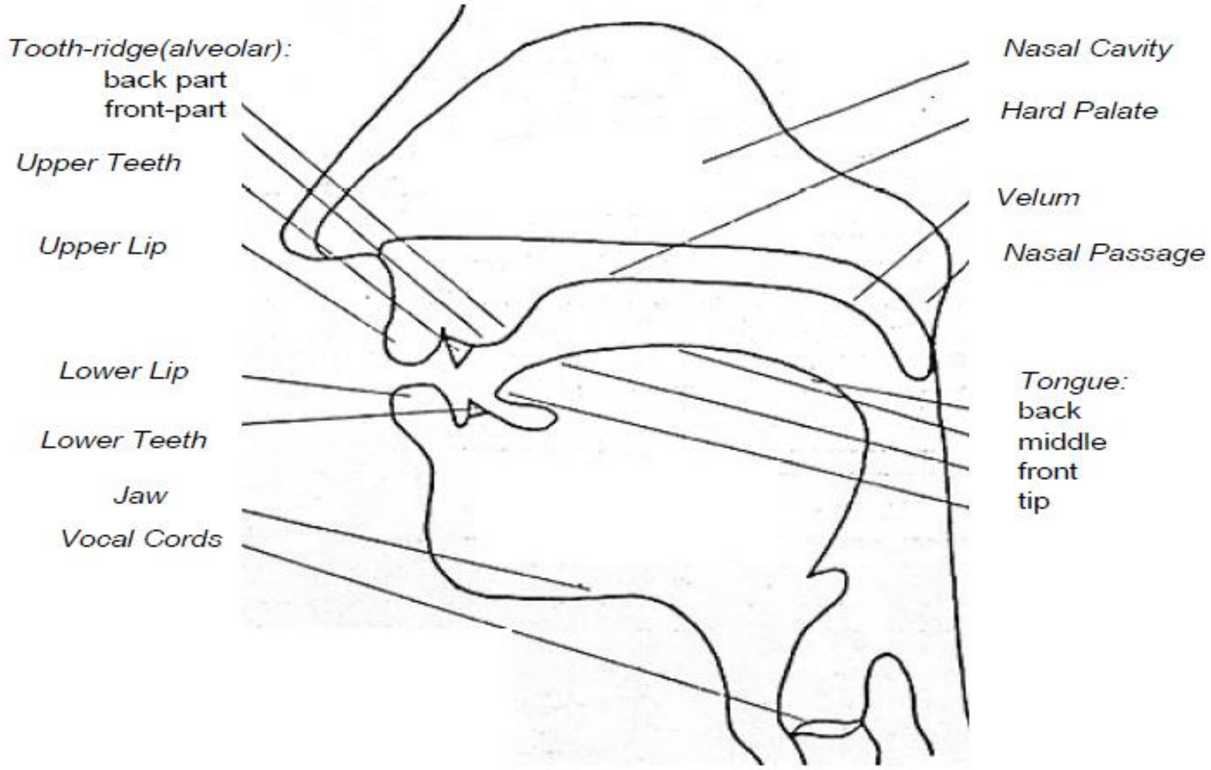


Figure (2.1): A schematic diagram of the human speech production apparatus. [14]

2.2.4 Production and classification of speech signal

"The speech sounds consist of three categories:

- Periodic
- Noisy
- Impulsive

Samples of speech Sounds generated with each of these three categories are shown in the word "shop," where the "sh," "o," and "p" are generated from a noisy, periodic, and impulsive source, respectively. The reader should speak the word "shop" slowly and determine where each sound source is occurring, i.e., at the larynx or at a constriction within the vocal tract" [4].

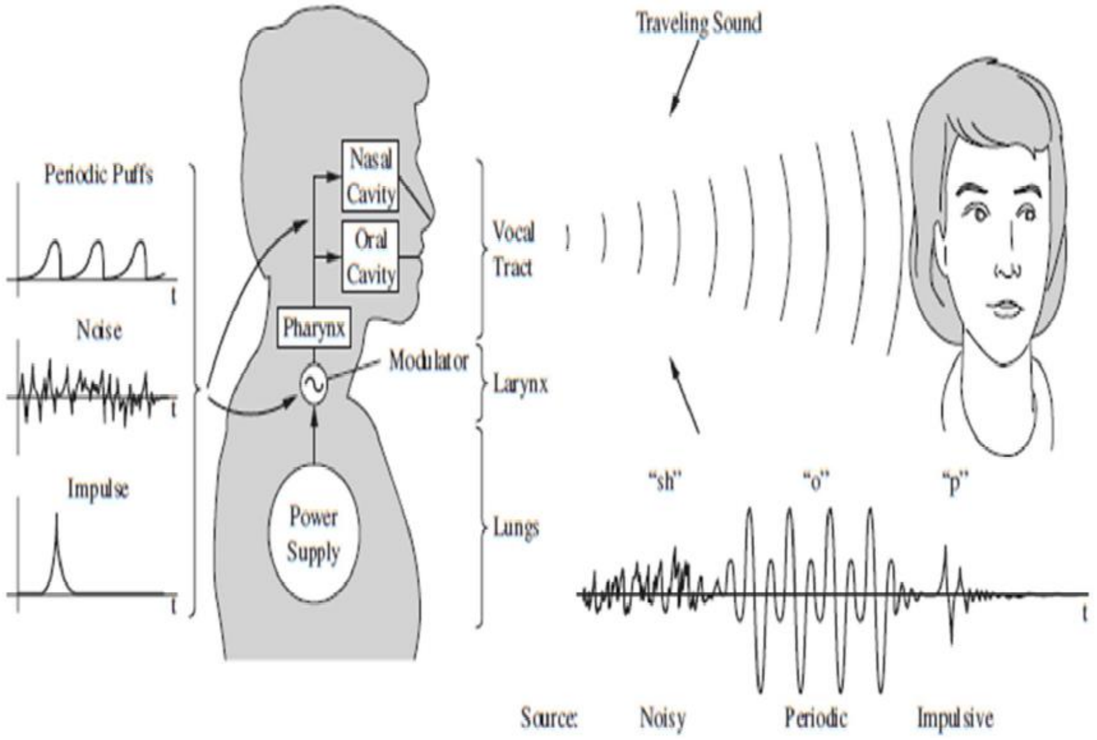


Figure (2.2): Simple view of speech production [4].

2.2.4.1 The Voicing Mechanism

The sound speech is divided into the voiced/voiceless distinction**.**

Voiced sounds, including vowels have regularly frequency when it pronounced, and voiceless sounds, such as consonants when it pronounced only air comes out and no frequencies [14].

Voiced like: "ز" "ط" "ض" "ص" "ا"

Voiceless like: "ت" "س" "ف" "ه"

The sound frequency vibrate is different between large man is 60 cycles per second (Hz), and 300 Hz or higher for a small woman or child [14].

This voicing mechanism is illustrated for sees in figure (2.3) showing both voiced and unvoiced partitions of the phenome.

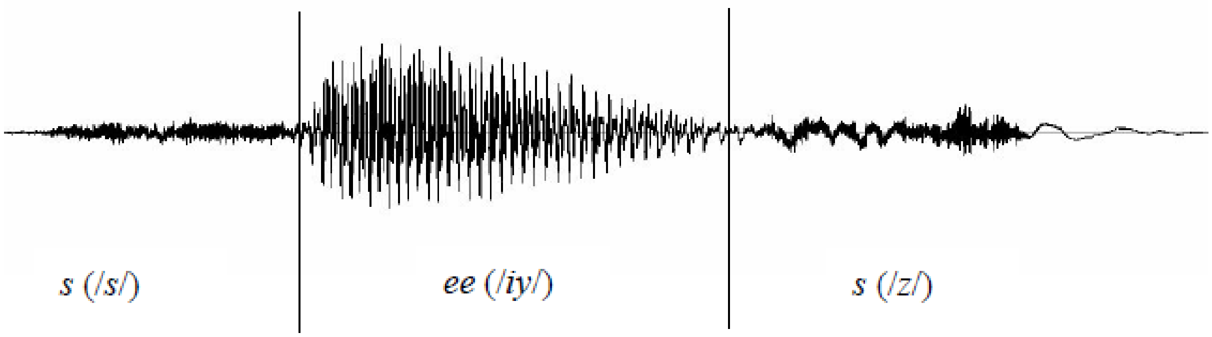


Figure (2.3): Waveform of sees, showing a voiceless phoneme /s, followed by a voiced sound, the vowel /iy/. The final sound, /z/, is a type of voiced consonant [14].

2.2.4.2 Vocal tract

"The vocal tract is comprised of the oral cavity from the larynx to the lips and the nasal passage that is coupled to the oral tract by way of the velum. The oral tract takes on many different lengths and cross-sections by moving the tongue, teeth, lips, and jaw and has an average length of 17 cm in a typical adult male and shorter for females, and a spatially-varying cross section of up to 20 cm^2. The pressure wave at the output of the vocal folds is heard during voicing simply a time-varying buzz-like sound which is not very interesting as illustrated in figure (2.4)" [4].

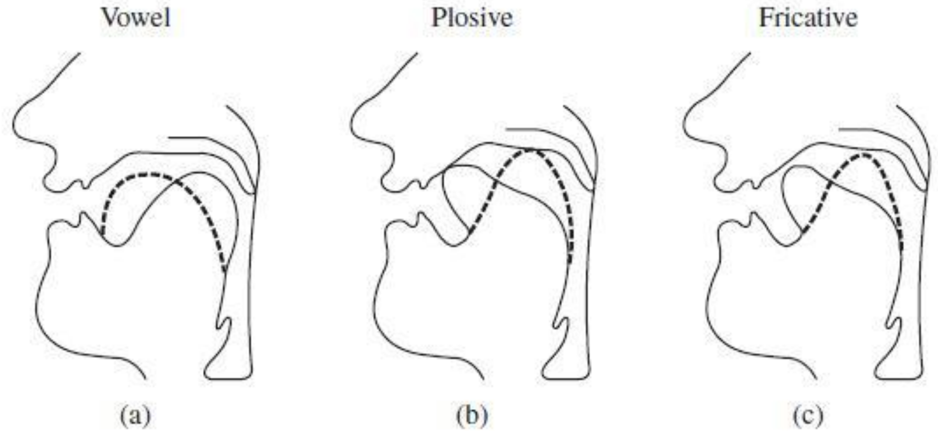


Figure (2.4): Illustration of changing vocal tract shapes for (a) vowels (having a periodic source), (b) plosives (having an impulsive source), and (c) fricatives (having noise source) [4].

2.2.5 Categorization of sound by source

Speech sounds can classes into different sources to the vocal tract; Speech sounds generated with vibration frequency are called voiced; likewise, sounds not so generated are called unvoiced. There are a variety of unvoiced sounds:

- Semi vowels
- Nasal consonants
- **Plosive consonants** (Stops): An example of a plosive is the “ك ج ت د ب”, A second unvoiced sound class is plosives created with an impulsive source within the oral tract.
- **Fricatives**: An example of frication is in the sound “ح هـ ز س” [4].

2.2.5.1 Placing cords in breathing state

“Cords open remarkably allowing the passage of air through them without any objection and offset this case the so-called whisper and external voices are called Voice Less Sounds like Alta, Althae, ha and kha .. etc. (التاء والثاء والحاء والخاء .. الخ) “[4].

2.2.6 Spectrographic analysis of speech

“There are two kinds of spectrograms: narrowband which gives good spectral resolution, e.g. a good view of the frequency content of sine waves with closely spaced frequencies, and wideband which gives good temporal resolution, e.g. a good view of the temporal content of impulses closely spaced in time. The difference between the narrowband and wideband spectrogram is the length of the window w[n, τ] for voiced speech” [4].

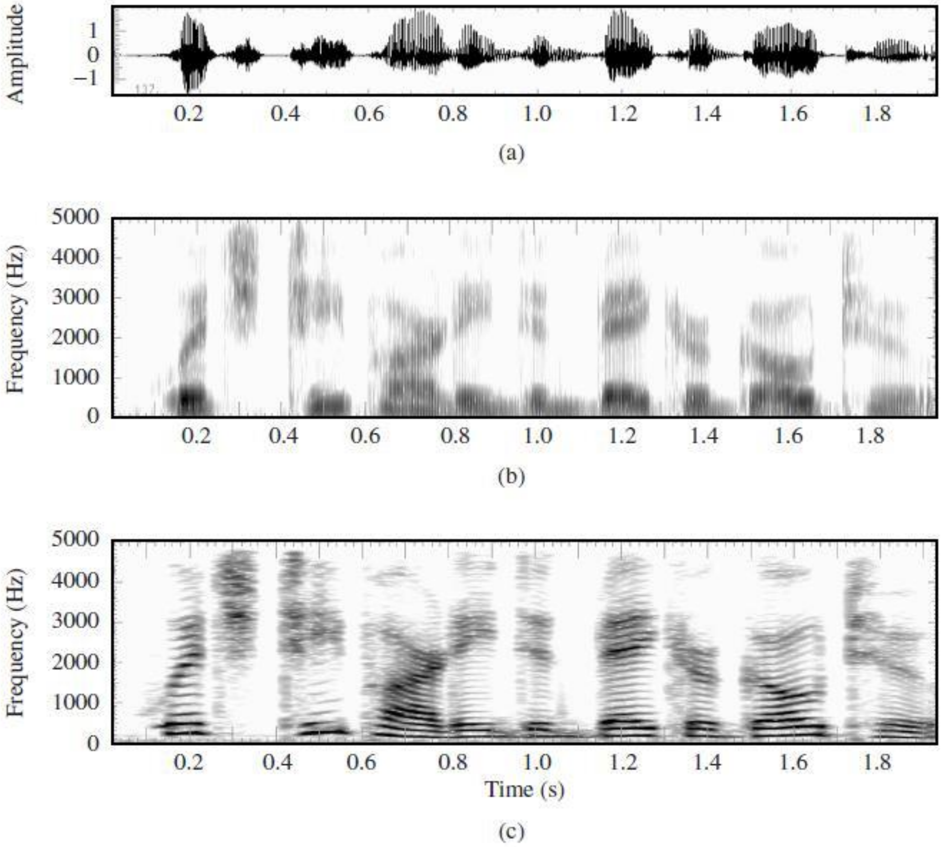


Figure (2.5): Comparison of measured spectrograms
(a) Speech waveform; (b) wideband spectrogram; (c) narrowband spectrogram [4]

2.2.7 Categorization of speech sounds

"Sound source can be created by either the vocal folds or with a constriction in the vocal tract. Speech sounds are studied and classified from the following perspectives:

1) The nature of the source: periodic, noisy, or impulsive, and combinations of the three.
2) The shape of the vocal tract.

3) The time-domain waveform, which gives the pressure change with time at the lips output.

4) The time-varying spectral characteristics revealed through the spectrogram.” [4]

2.2.8 Elements of language

To distinctive of speech sound used *phoneme* unit, its represent the features of word like letters and phoneme classes. for example, the words “cat,” “bat,” and “hat” consist of three speech sounds, the first of which gives each word its distinctive meaning, being from different phoneme classes. Different languages contain different phoneme sets [4].

2.2.8.1 Vowels

“Vowels contain three subgroups defined by the tongue hump being along the front, central, or back part of the palate.

It is the second major section of the votes of the language, in the language movements differ from environment to another is because of this difference to the habits pronunciation local dialect, put lips is the general standard, which is classified Ken types of movements or patterns, movements term that was called in the old on the fatha and damma and Kasra or what is known as short vowels, where long vowels are الالف والواو والياء sleepless movements a:” (ـــَــِــُـ) [4].

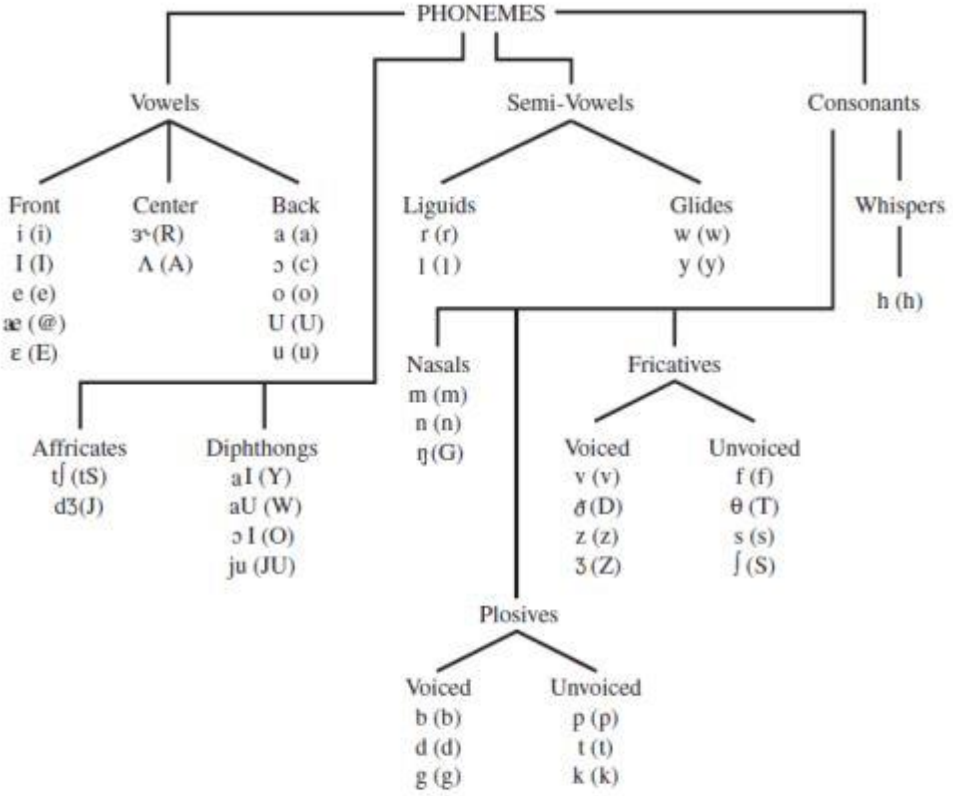


Figure (2.6): Phonemes in American English [4]

2.2.8.2 Consonants

“Consonant identification depends on a number of factors including the formants of the consonant, formant transitions into the formants of the following vowel, the voicing (or unvoicing) of the vocal folds during or near the consonant production “[4].

2.2.8.2.1 Nasals

"The second large phoneme grouping is that of consonants. The consonants contain a number of subgroups: nasals, fricatives, plosives, whispers, and affricates. We begin with the nasals since they are closest to the vowels. In Arabic language م ,ن
Source: As with vowels, the source is quasi-periodic airflow puffs from the vibrating vocal folds [4].

System: The velum is lowered and the air flows mainly through the nasal cavity, the oral tract being constricted” [4].

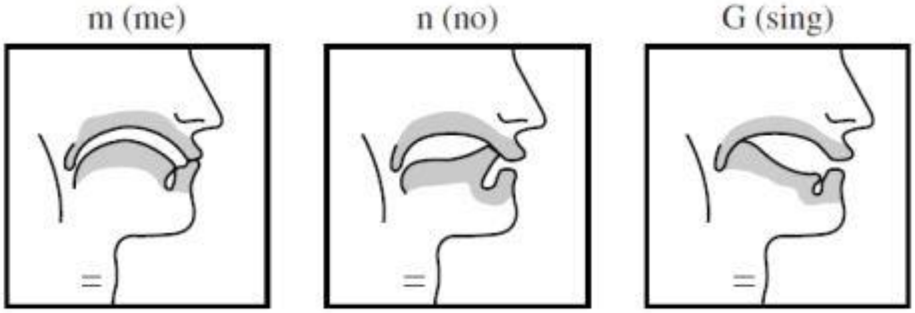


Figure (2.7): Vocal tract configurations for nasal consonants. Oral tract constrictions occur at the lips for /m/, with the tongue tip to the gum ridge for /n/, and with the tongue body against the palate near the velum for /ng/. Horizontal lines denote voicing [4].

2.2.8.2.2 Fricatives

“Fricative consonants are specified in two classes: voiced and unvoiced fricatives.

Source: In unvoiced fricatives, the vocal folds are relaxed and not vibrating.

System: The location of the constriction by the tongue at the back, center, or front of the oral tract, as well as at the teeth or lips, influences which fricative sound is produced. e.g. in Arabic الهاء ، العين ، الحاء ، الغين ، الخاء ، الشين ، الصاد ، السين ، الزاى ، الظاء الذال ، الثاء والفاء “[4].

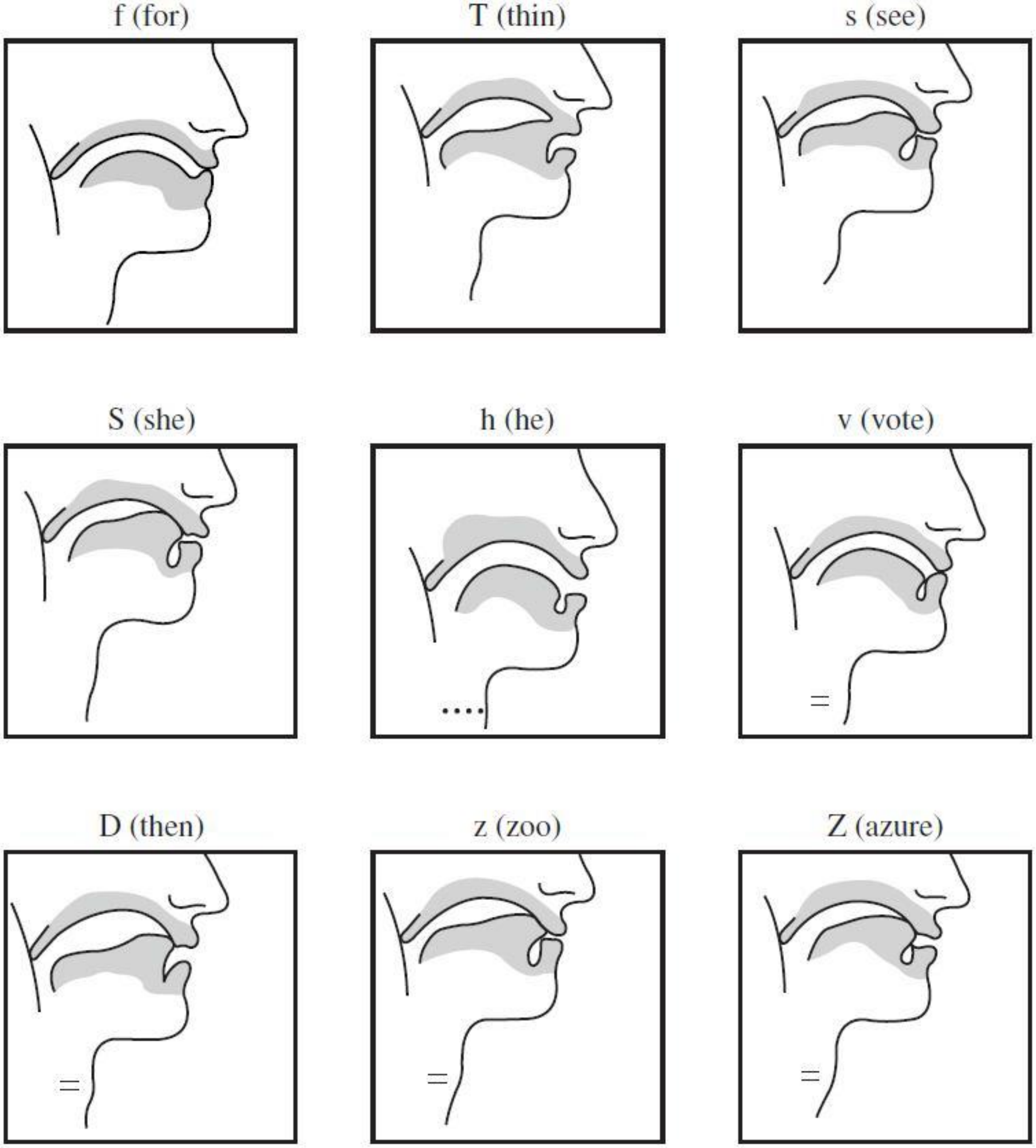


Figure (2.8): Vocal tract configurations for pairs of voiced and unvoiced fricatives. Horizontal lines denote voicing and dots denote aspiration [4].

2.2.8.2.3 Whisper

“The whisper is a consonant similar in formation to the unvoiced fricative; we place the whisper in its own consonantal class. We saw earlier that with a whisper the glottis is open and there is no vocal fold vibration. An example is /h/.” [4].

2.2.8.2.4 Plosives

“As with fricatives, plosives are both unvoiced and voiced.

Source and System: With unvoiced plosives, a “burst” is generated at the release of the buildup of pressure behind a total constriction in the oral tract. E.g. in Arabic همزة ، القاف ، الكاف ، الدال ، الضاد ، التاء ، الطاء “[4].

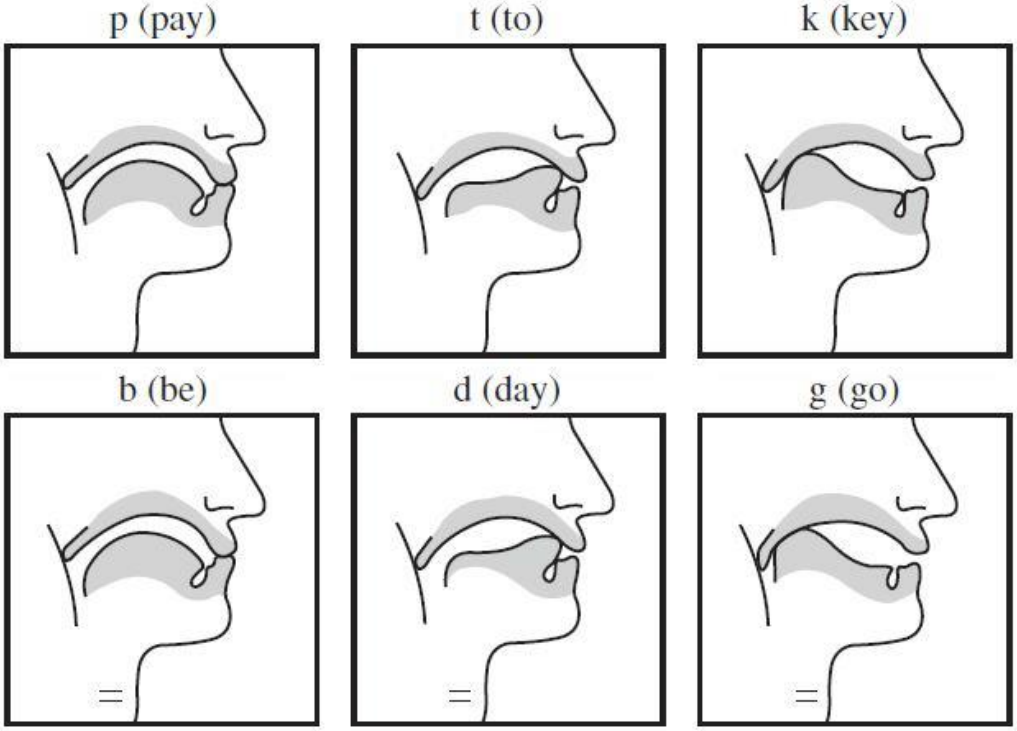


Figure (2.9): Vocal tract configurations for unvoiced and voiced plosive pairs. Horizontal lines denote voicing [4].

2.2.8.3 Semi-Vowels:

“This class is also vowel-like in nature with vibrating folds. There are two categories of semi-vowels: glides (/w/ as in “we” and /y/ as in “you”) and liquids (/r/ as in “read” and /l/ as in “let”). In Arabic is و, ي “[4].

2.2.9 Feature extraction module

The goal of feature extraction is to identify the basic information of speach signal that represent the properties of it. There are three main types of feature extraction techniques, namely linear predictive coding (LPC), Mel frequency cepstrum coefficient (MFCC) and perceptual linear prediction (PLP). MFCC and PLP are the most commonly used feature extraction techniques in modern ASR systems [11].

The main part of the feature extraction lies the short-term spectral analysis (e.g. discrete Fourier transform). The basic process here is to extract a sequence of features for each short-time frame of the input signal, with an assumption that such a small segment of speech is sufficiently stationary to allow meaningful modeling. The efficiency of this phase is important for the next phase since it affects the behavior of modeling process [11]. I used the Mel-frequency Cepstral Coefficients (MFCC) for feature extraction.

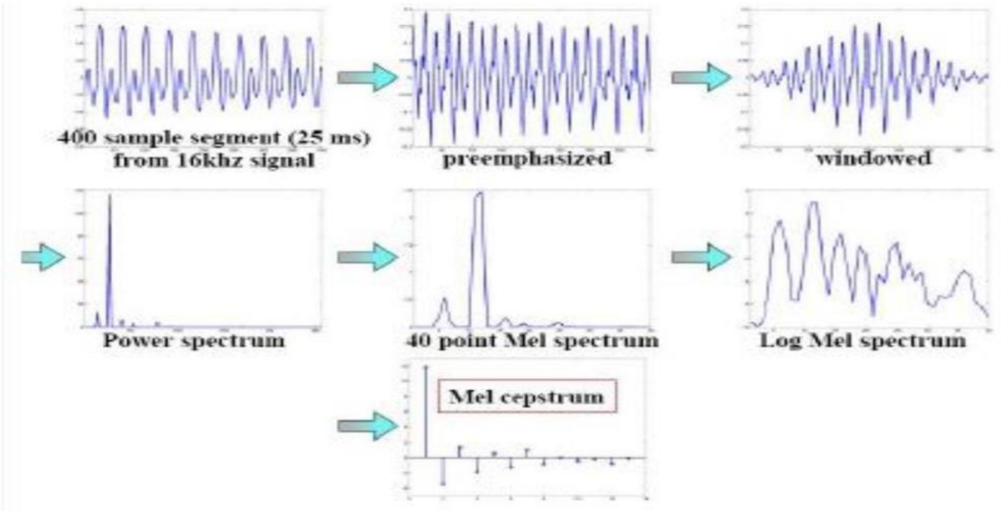


Fig (2.10): Extraction of MFCC Feature for a Frame [4]

2.2.10 Spectral analysis

Spectral analysis is concerned with determining the frequency content of an arbitrary signal. Feature extraction is done on short time basis. The speech signal is divided into overlapped fixed length frames. A set of cepstrums domain or frequency domain parameters, called feature vector derived from each frame. Different signal processing operations such as pre-emphasis, framing, windowing and Mel cepstrum analysis performed on the input signal, at different stages of the MFCC algorithm [11].

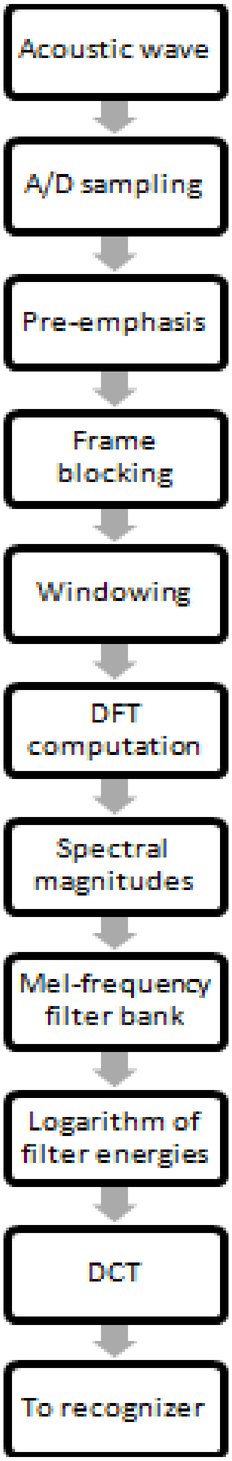


Figure (2.11): Block diagram of the feature extraction processing [11].

2.2.10.1 Pre-emphasis

Noise has a greater effect on the higher modulating frequencies than the lower ones. Hence, higher frequencies artificially boosted to increase the signal-to-noise ratio. Pre-emphasis process performs spectral flattening using a first order finite impulse response (FIR) filter. Equation (2.1) represents first order FIR filter [11].

$$H(z) = 1 - \alpha z^{-1}, 0.9 \le \alpha \le 1.0 \quad \ldots\ldots \quad (2.1)$$

2.2.10.2 Frame Blocking

Speech is a non-stationary signal. If the frame is too long, signal properties may change too much across the window, affecting the time resolution adversely. If the frame is too short, resolution of narrow-band components will be sacrificed, affecting the frequency resolution adversely [11].

There is a trade-off between time resolution and frequency resolution. Overlapping frames are used to capture information that may occur at the frame boundaries. Number of frames is obtained by dividing the total number of samples in the input speech file. For covering all samples of input, last frame may require zero padding. All frames are stored as rows in one single matrix with number of rows equal to number of frames and number of columns, which is also equal to the frame width [11].

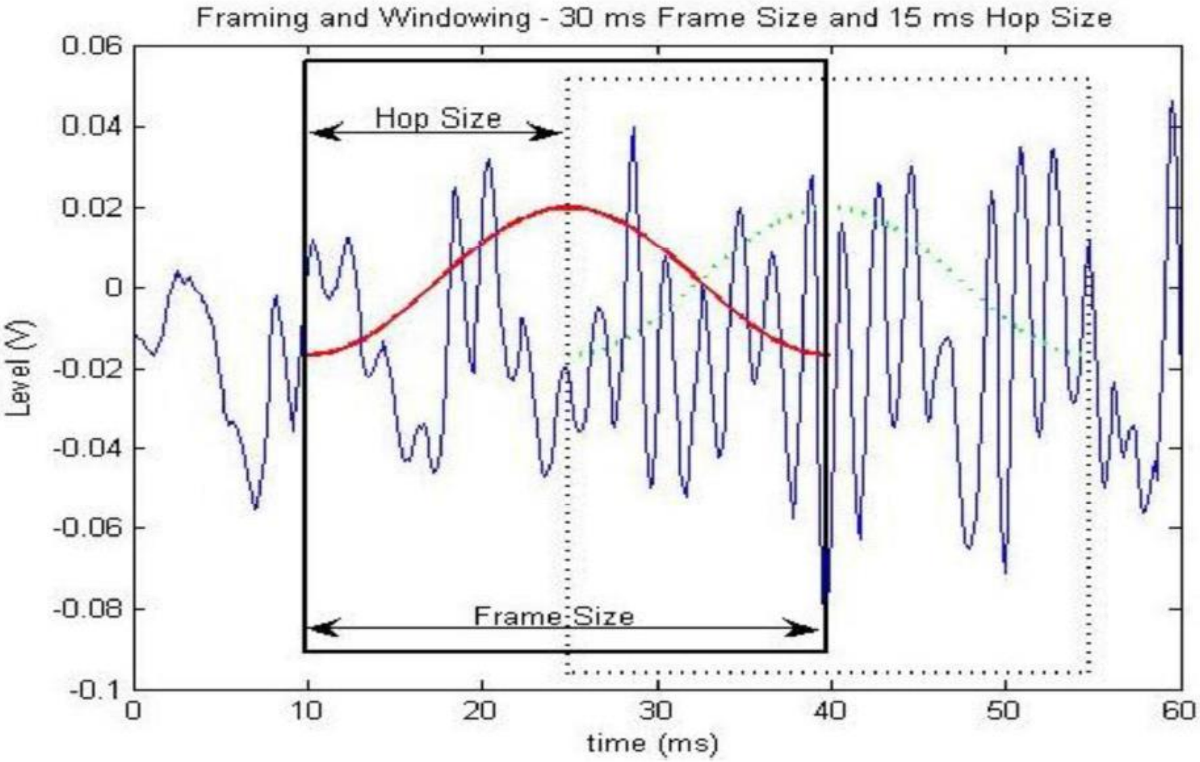


Fig (2.12): Framing and overlapping [4]

2.2.10.3 Windowing

Discontinuities at the beginning and end of the frame are likely to introduce undesirable effects in the frequency response. Hence, each row is multiplied by window function. A window alters the signal, tapering it to nearly zero at the beginning and the end. Hamming window introduces the least amount of distortion. Equation (2.2) shows the discrete time domain representation of Hamming window function.

$$h[\mathrm{n}] = \begin{cases} 0.54 - 0.46\cos(2\pi n / N), 0 \le n \le N \\ 0, otherwise \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.2)$$

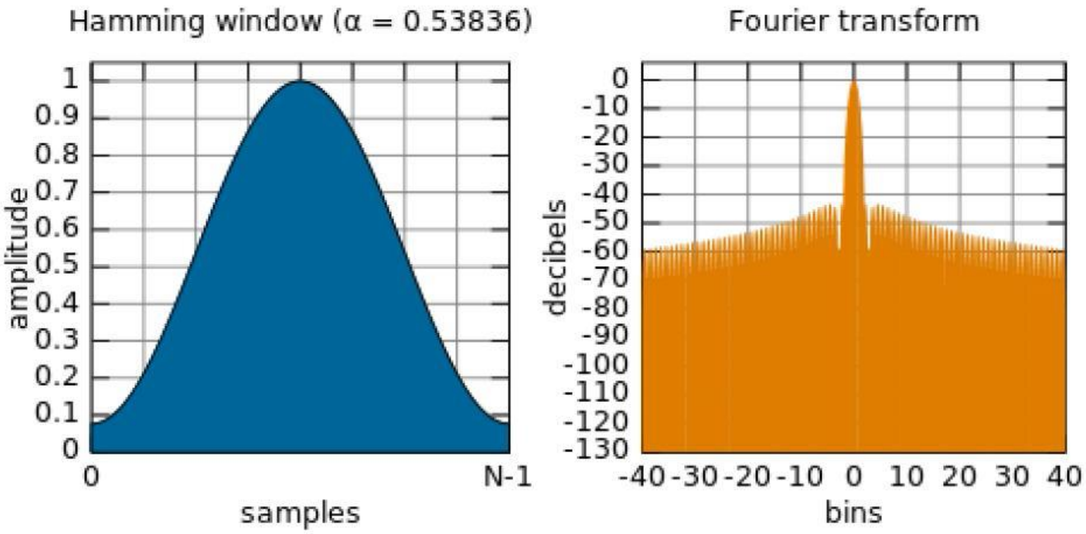


Fig (2.13): Hamming window and its frequency spectrum [4]

2.2.10.4 Spectral magnitude of DFT

Spectral information means the energy levels at different frequencies in the given window. Time domain data is converted into frequency domain to obtain the spectral information. Time domain data is converted to frequency domain by applying Discrete Fourier Transform (DFT) on it [11]. Equation (2.3) represents DFT.

$$X(\mathrm{k}) = \sum_{n=0}^{N-1} x(\mathrm{n}) e^{-j2\pi kn/N}, 0 \le k \le N-1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

Here, x(n) represents input frame of N samples and X(k) represents its equivalent DFT. We use 256-point FFT algorithm to convert each frame of 256 samples into its equivalent DFT. FFT output is a set of complex numbers i.e. real and imaginary parts. Speech recognition systems deal with real data. Hence, complex value is always ignored [11]. If we assume the real and imaginary parts of X(k) as Re(X(k)) and Im(X(k)), then the spectral magnitude of the speech signal can be obtained by using equation (2.4). Spectral magnitudes of each frame are stored as rows in one single matrix with number of rows equal to number of frames and number of columns equal to 256, which is also equal to the frame width.

$$|X(\mathrm{k})| = \sqrt{(\mathrm{Re}(\mathrm{X}(\mathrm{k})))^2 + (\mathrm{Im}(\mathrm{X}(\mathrm{k})))^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2.4)}$$

2.2.10.5 Mel frequency filter bank

Mel-frequency analysis of speech is based on human perception experiments. It has been proved that human ears are more sensitive and have higher resolution to low frequency compared to high frequency. Hence, the filter bank is designed to emphasize the low frequency over the high frequency. Also the voice signal does not follow the linear frequency scale used in FFT. Hence, a perceptual scale of pitches equal in distance, namely Mel scale is used for feature extraction. Mel scale frequency is proportional to the logarithm of the linear frequency, reflecting the human perception [11]. Figure (2.14) shows frequencies in Mel scale plotted against frequencies in linear scale. Equation (2.5) is used to convert linear scale frequency into Mel scale frequency.

$$Mel(\mathrm{f}) = 2595\log_{10}\left(1 + \frac{f}{700}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2.5)}$$

Triangular band pass filters are used to extract the spectral envelope, which is constituted by dominant frequency components in the speech signal. Thus, Mel-frequency filters are triangular band pass filters non-uniformly spaced on the linear frequency axis and uniformly spaced on the Mel frequency axis, with more number of filters in the low frequency region and less number of filters in the high frequency region [11].

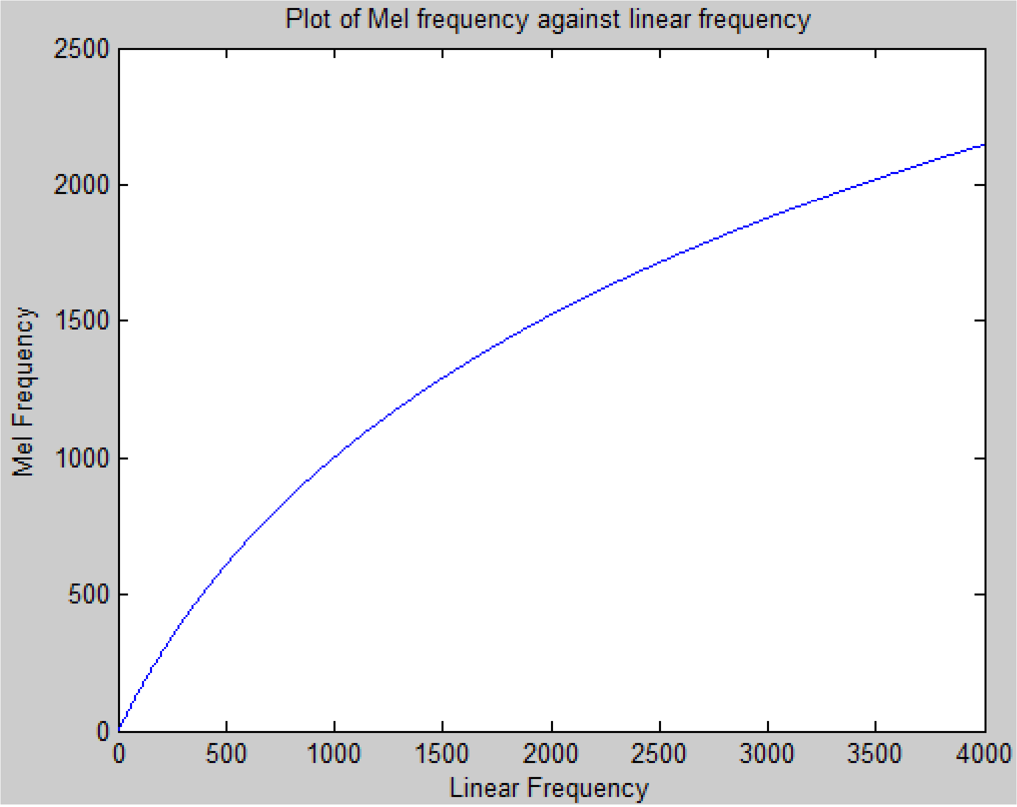


Figure (2.14): Plot of Mel frequencies against linear frequencies [11]

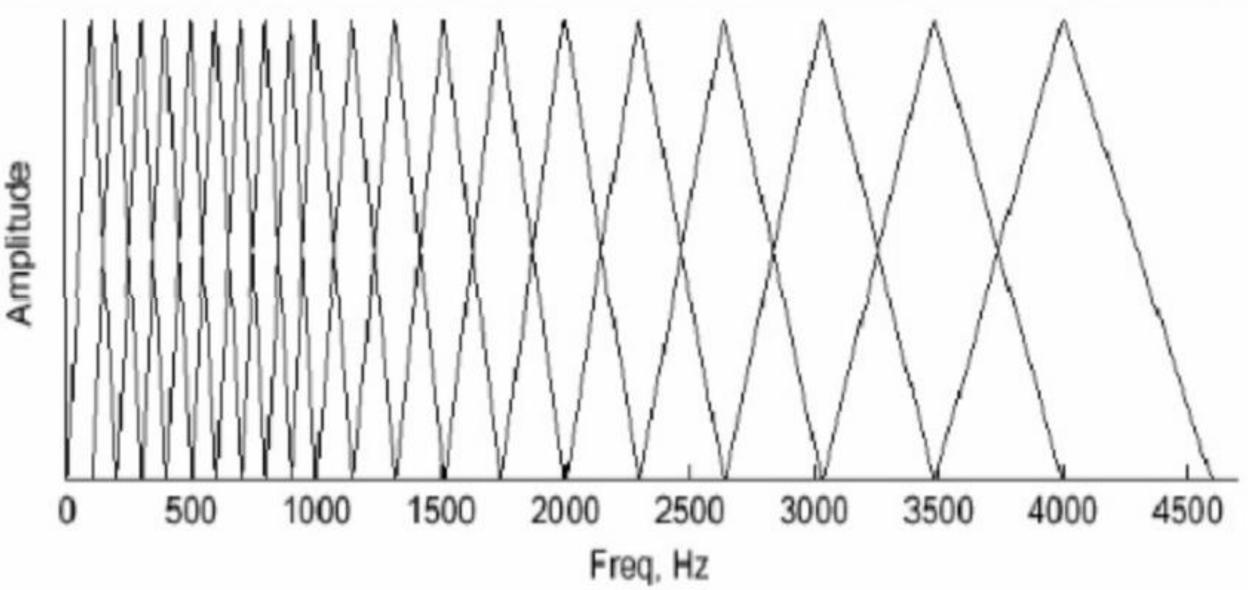


Fig (2.15): Filter bank in Mel frequency scale [4]

From figure (2.15), magnitude response of each filter is equal to unity at the center and decreases linearly to zero at the center frequencies of two adjacent filters. No frequency resolution required to put the filters at the exact linear frequency points, since we are processing the data in discrete frequency domain. Hence, we round these linear frequencies to nearest FFT points [11].

It was observed that as m increases, the difference between centers of two adjacent filters increases in linear scale and remains the same in Mel scale. Equation (2.6) represents the filter bank with M (m = 1, 2, 3….M) filters, where m is the number of triangular filter in the filter bank [11].

$$Hm(\mathrm{k}) = \begin{cases} 0, \text{for } k < f(\mathrm{m}-1) \\ \dfrac{k - f(\mathrm{m}-1)}{\mathrm{f(m)} - \mathrm{f(m}-1)}, \text{for } f(\mathrm{m}-1) \leq \mathrm{k} \leq f(\mathrm{m}) \\ \dfrac{f(\mathrm{m}+1) - \mathrm{k}}{f(\mathrm{m}+1) - \mathrm{f(m)}}, \text{for } f(\mathrm{m}) \leq \mathrm{k} \leq \mathrm{f(m}+1) \\ 0, \text{for } k > f(\mathrm{m}+1) \end{cases} \quad \text{……………..…... (2.6)}$$

Each triangular filter in the filter bank satisfies equation (2.7) [11].

$$\sum_{m=0}^{M-1} Hm(\mathrm{k}) = 1 \quad \text{…………………………………………………..…….. (2.7)}$$

Mel frequency filter bank applied to every N samples frame and the filtered response computed. In frequency domain, filtering is obtained by multiplying the FFT of signal and transfer function of the filter on element by element basis [11].

2.2.10.6 Logarithm of filter energies

Human ears smooth the spectrum and use the logarithmic scale approximately. We use equation (2.8) to compute the log-energy i.e. logarithm of the sum of filtered components for each filter.

$$S(\mathrm{m}) = \log_{10}\left[\sum_{k=0}^{N-1}\left|X(\mathrm{k})\right|^{2}.Hm(\mathrm{k})\right], 0 \le m \le M \quad \text{………………………... (2.8)}$$

Thus, each bin per frame per filter holds the log-energy obtained by computing logarithm of weighted sum of spectral magnitudes in that filter-bank channel. Output of this stage is stored in a matrix with number of rows equal to number of frames and number of columns equal to number of filters in the filter bank [11].

2.3 Related Works

2.3.1 Arabic Speech Recognition Using MFCC Feature Extraction and ANN Classification

Elvira Sukma Wahyuni, (2017), recognized three spoken Arabic letters of hijaiyah having the same pronunciation by Indonesian speakers but has different makhraj in Arabic letters of sa, sya and tsa. Mel-Frequency Cepstral Coefficients (MFCC) based feature extraction method and Artificial Neural Network (ANN) were used for classification. The result was average accuracy of 92.42%, and each letters (sa, sya, and tsa) has accuracy of 92.38%, 93.26% and 91.63% respectively [2].

2.3.2 Voice Recognition by using Machine Learning A Case Study of some Rules of Tajweed

Safaa Omer Mohammed Nssr, (2016), designed a model for classification four main rules of (altajweed) of the Allah name (mofakham, morakaq), moony, and sunny (لام). LPC (liner predictive coding) and MFCC (Mel frequency cepstrum) were applied to extract audio features and being classified by neural networks and hidden Markov model (HMM) for different readers (males) and (female). The results obtained by HMM have accuracy of 90% with Allah (moufakhum), 92% with Allah (mourqeq), 83.3% with sunny (لام) and 80% with moony (لام) and the results obtained by neural networks reached high accuracy of 95% with Allah (moufakhum), 94% with Allah (mourqeq) , 93% with moony (لام) and 92.3% with sunny (لام) [4].

2.3.3 Development of Quran Reciter Identification System Using MFCC and Neural Network

Tayseer Mohammed Hasan Asda, Teddy Surya Gunawan, Mira Kartiwi, Hasmah Mansor, (2016), developed Quran reciter recognition and identification system based on mel frequency cepstral coefficient (MFCC) feature extraction and artificial neural network (ANN). A database of five Quran reciters created, trained and tested. Accuracy was 91.2% [6].

2.3.4 Phonetic Recognition of Arabic Alphabet letters using Neural Networks

Moaz Abdulfattah Ahmad, Rasheed M. ElAwady, (2011), they designed a system to recognize the Arabic Alphabet letters that convert the spoken words into written text. The main features spoken signal were extracted using Principal Component Analysis (PCA) technique. An accuracy of 96% has been achieved over large dataset [7].

2.3.5 Speech and Language Recognition using MFCC and DELTA-MFCC

Samiksha Sharma, Anupam Shukla and Pankaj Mishra, (2016), designed a model to recognize speech and language for countries that have many languages. Four languages Hindi, English, Sanskrit and Telugu have been used. MFCC and delta-MFCCs were applied for feature extraction and ANN was used to model and classify the input as a set of 18 words and languages. In first experiment, the overall sound recognizer’s performance was 83.89% and language recognizer’s performance was 83.3%. In second experiment, radial basis function network used as classifier and overall word recognition accuracy was 91.7% and language recognition accuracy was 91.1% [8].

2.3.6 Speech Recognition using Neural Network

Pankaj Rani, Sushil Kakkar, and Shweta Rani, (2015), presented back propagation algorithm using neural network. Voices of different persons of various ages were recorded in high quite environment with high quality. The features of the recorded samples were extracted by using LPC algorithm. The best training performance rate was 2.2596-20 at epoch 4 [9].

2.3.7 Arabic Phoneme Recognition Using Neural Networks

Mansl El-Obaid, Amer Al-Nassiri and Iman Abuel Maaly, (2006), designed a system for recognition of Arabic speech phonemes using artificial neural networks. Three techniques were used as follow: First, the preprocessing in which the original speech is transformed into digital form using FIR filter and Normalization. Second, using Cepstral coefficients, with frame size of 512 samples, 170 overlapping, and hamming window. Finally, using Multi-Layer Perceptron Neural Network (MLP), based on Feed Forward Back propagation. The results achieved was within 96.3% for most of the 34 phonemes [10].

CHAPTER THREE: DESIGN OF PROPOSED VOICE SYSTEM

3.1 Introduction

This chapter explains experimental steps of the research. It demonstrates the research project implementation consisting of three phases: First phase is the digital signal processing, the second phase is feature extraction and the third phase is the machine learning for classification of which the correct letter has spoken.

3.2 MATLAB Platform

MATLAB is one of a number of commercially available, sophisticated mathematical computation tools. They differ in the way they handle symbolic calculations and more complicated mathematical processes, such as matrix manipulation. MATLAB (short for **Mat**rix **Lab**oratory) excels at computations involving matrices. In fact, MATLAB was originally written in FORTRAN and later rewritten in C, a precursor of C++. MATLAB is optimized for matrices. MATLAB executes faster than a similar program in a high-level language [12].

This project used this MATLAB software to extract the features of letter voices using MFCC (Mel Frequency Cepstral Coefficient) coefficients, train and then finally classify using neural networks and check the performance of the system.

3.3 Framework of the System

The implemented system consists of the following stages as shown in figure (3.1):

1. Silence removal (Segmentation).
2. Digital signal pre-processing techniques including:

   2.1 Doing frame blocking (Divide signal into frames).

   2.2 Windowing (The amplitude spectrum) and filtering.

   2.3 The frequency domain analysis. (Using Fast Fourier Transform (FFT)).
3. Mel- Frequency Warping (convert to mel spectrum).
4. Cepstrum (take the discrete cosine transform).
5. Classification (Using neural networks).

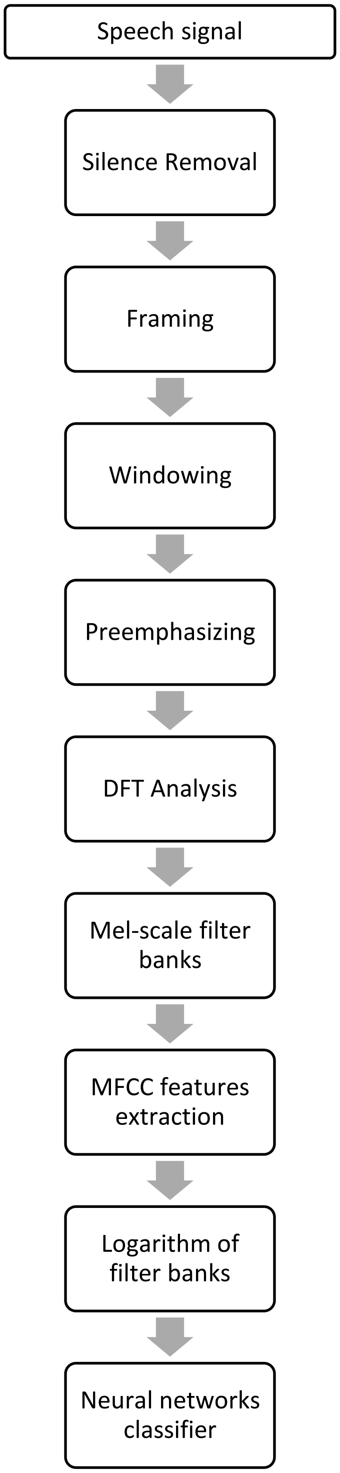


Figure (3.1): Framework of the project

3.4 Design and Implementation of the System

3.4.1 Phase one (Digital signal pre-processing):

3.4.1.1 Step one (Reading the voice signal):

At the beginning and for designing the system, photonic Arabic letters database has been loaded from (Lebanese Arabic from Scratch - Alphabet Book).

This database contains 10 different alphabetic Arabic letters, 6 vowels for each, with high quality voice recorded, each of which with sampling frequency of 44000 Hz and voice clip duration of 2 seconds. The total database contains 60 different files, so the overall examples for training the neural network are 3600 samples of frames (60 frames each). The letters used are: Ain (ع), Gin (غ), Taa (ت), Tta (ط), Haa (ح), Kha (خ), Sud (ص), Daa (ض), Dha (ظ) and Tha (ذ), as shown in table (3.1) below:

Table (3.1): Database of ten letters and six vowels each

No. of the letter	Letter	Vowel
1	Ain (ع)	عَ
2	Ain (ع)	عِ
3	Ain (ع)	عُ
4	Ain (ع)	عٌ
5	Ain (ع)	عًا
6	Ain (ع)	عٍـ
7	Daa (ض)	ضَ
8	Daa (ض)	ضُ
9	Daa (ض)	ضِ
10	Daa (ض)	ضًا
11	Daa (ض)	ضٌ
12	Daa (ض)	ضٍ

No. of the letter	Letter	Vowel
13	Dha (ظ)	ظَ
14	Dha (ظ)	ظُ
15	Dha (ظ)	ظِ
16	Dha (ظ)	ظًا
17	Dha (ظ)	ظٍ
18	Dha (ظ)	ظٌ
19	Gin (غ)	غَ
20	Gin (غ)	غِ
21	Gin (غ)	غُ
22	Gin (غ)	غًا
23	Gin (غ)	غٌ
24	Gin (غ)	غٍ
25	Haa (ح)	حَ
26	Haa (ح)	حُ
27	Haa (ح)	حِ
28	Haa (ح)	حاً
29	Haa (ح)	حٌ
30	Haa (ح)	حٍ
31	Kha (خ)	خَ
32	Kha (خ)	خُ
33	Kha (خ)	خِ
34	Kha (خ)	خاً
35	Kha (خ)	خٌ
36	Kha (خ)	خٍ

No. of the letter	Letter	Vowel
37	Sud (ص)	صَ
38	Sud (ص)	صِ
39	Sud (ص)	صُ
40	Sud (ص)	صاً
41	Sud (ص)	صٌ
42	Sud (ص)	صٍ
43	Taa (ت)	تَ
44	Taa (ت)	تُ
45	Taa (ت)	تِ
46	Taa (ت)	تاً
47	Taa (ت)	تٌ
48	Taa (ت)	تٍ
49	Tta (ط)	طَ
50	Tta (ط)	طُ
51	Tta (ط)	طِ
52	Tta (ط)	طاً
53	Tta (ط)	طٌ
54	Tta (ط)	طٍ
55	Tha (ذ)	ذَ
56	Tha (ذ)	ذُ
57	Tha (ذ)	ذِ
58	Tha (ذ)	ذاً
59	Tha (ذ)	ذٌ

No. of the letter	Letter	Vowel
60	Tha (ذ)	ذٍ

3.4.1.2 Step two (Extracting the voice data):

The system was designed by MATLAB and the database was loaded completely on the current directory. For purpose of demonstrating the design and test of both phase one and phase two, just one letter was taken for this purpose (Ka’af) letter from one male and the rest of letters was applied later in phase three.

After applying just one letter, it was loaded, played (for making sure that it is the true voice) and the data was finally extracted from this voice clip and then was plotted.

3.4.1.3 Step three (Segmentation):

The loaded and sampled signal was subjected to pre-processing techniques based on the principles of digital signal processing including: silence removal, framing, windowing and filtering.

To process analog signals it is first necessary to convert them into digital form which is called analog-to-digital (A/D) conversion. For the purpose of recognition, speech needs to be segmented into phonetic units.

The segmentation of speech was depending on two methods, manual and automatic segmentation. In this project auto segmentation was used.

It was noticeable that all sounds do not start from sample 1, although most of them are posted in the beginning of the carrier. The reason is that there was silence in the beginning of each sound.

Each voice letter clip contains two parts: voiced part and unvoiced part which is called silent part.

This silent part on either sides of the centered voiced block in the original signal was considered as noise and contains no useful information, so it has to be removed.

There are many techniques commonly used for silence removal and one of these used in this project was STE method (Short Term Energy).

In MATLAB, STE was first calculated using *for* loop for specific each frame and then it was normalized.

```
for i=1:r
    ste(i)=sum(frames(i,:).^2);
end
```

Secondly, it was calculated again for all the frames of the original signal.

Then, it was plotted together in on figure with the original signal to check accurately at which minimum amplitude of the silence part that has to be removed by assigning STE for it.

Finally, STE threshold value was determined and then scanning all frames of the original signal to check this value and extracting the regions of silence in original signal and the silence was removed on both sides of voice carrier.

3.4.1.4 Step four (Frame blocking, windowing, filtering and DFT analysis)

3.4.1.4.1 Frame blocking

Speech signal is a continuous signal and it was divided into frames for proper analysis [13]. The original speech signal (after silence removal) was named as SR. signal for concise purpose.

SR. signal was divided into frames, each one has a duration of 0.02 seconds (20 msec).

To get the total number of samples per frame (named frame size), the frame duration was multiplied by the sampling frequency.

To get total no. of frames in the SR. signal, the total number of samples in the signal (n) was divided by frame size.

The SR. signal was framed using *for* loop in MATLAB to enable choosing appropriate or specific frame.

3.4.1.4.2 Windowing and filtering

There are many types of window functions used in digital FIR filtering (Finite Impulse Response) and one of these selected here was Hamming window which multiplied by the chosen frame and then filtered it using high pass filter to specify only the useful (voiced) information block and isolate the low frequencies part which contains no useful information.

Hamming window was designed in MATLAB with a size similar to each frame size. The output of this window is a column vector and also the frame should be in equal dimension with the window.

The last problem was the matrix dimension between the window and frame which has been solved by specify the frame and transposing it as follow:

```
fr_win=frames(45,:)'.*hamming(length(frames(45,:)));
```

High pass filter (HPF) coefficients were selected to be [1 – 0.95] respectively.

The cutoff frequency was 5kHz.

It was implemented in MATALB using the following command:

`F=filter(alpha,1,fr_win);` where alpha is the filter coefficients, one for the amplitude which was in linear scale and finally the (fr_win) stands for windowed frame.

The filtered frame (F) was plotted and compared with the windowed frame in the discrete time domain. After this, FFT output was compared between them in the frequency domain.

Power Spectrum Density (PSD) analysis tool was also implemented and plotted in MATALB to show how the energy of the voice signal or frame is decreasing gradually as the frequency components increase. One of PSD types used in this project was `welch spectrum`.

3.4.1.4.3 Discrete Fourier Transform (DFT) analysis

The computed Discrete Fourier Transform (DFT) was applied for each step of the major phases of the system to analyze the frequency elements of processed signal and compare between the progressing steps.

The DFT was implemented in MATALB using the direct command: *fft*.

3.4.2 Phase two (Feature extraction):

Feature extraction is the process of extracting the features of a sound wave signal, which is basically the extraction of the fundamental parameters identifying a speech signal.

At the beginning, the linear frequency scale was converted into Mel scale and then the maximum number (M) of equal spaced frequency filter triangular banks was determined to be for 39 filter bands, starting from zero frequency and ending with a maximum frequency of approximately 500 Hz.

Because there were a problem and difficulty in equalizing the sampling frequency, frame size, frame duration and hence maximum number of frames between both the standard (or target) letter voice signals and the actual spoken signals, so this was solved by getting the filterbanks for all the frames in the voice signal.

FFT was applied for all frames in SR. signal; energy was calculated excluding the negative frequency part (the second half) then no. of periodograms (Normalized power) was also calculated to get the triangular shaped filterbanks written by the command below:

```
mfccShapes = triangularFilterShape(ff,N,fs);
```

The output of this function was mfcc matrix and so for easy analysis and classification later, it was converted into a column vector of 39 * 1 size.

Finally the logarithm of this function was also computed to get the equivalent cepstral coefficients which will be used in the next phase for purpose of training and classification.

3.4.3 Phase three (classification):

In this project, artificial neural networks have been designed, trained, validated and tested for three experiments.

3.4.3.1 First Experiment

In this experiment, only the neural networks was used for training one letter (consonant) without vowel.

Backpropagation neural networks was selected for this goal with one input layer (15 neurons equal to the number of cepstral coefficients), one hidden layer (100 elements) and one output layer (one node for one of either two numerical classes 0 standing for not the target letter and 1 standing for correct letter).

3.4.3.2 Second Experiment

In this experiment, the neural networks was used for training 10 letters with six vowels each. The total is 60 different vowels.

Backpropagation neural networks was selected for this goal with one input layer (39 neurons equal to the number of cepstral coefficients), one hidden layer (48 elements) and one output layer (60 nodes for 60 classes of different vowels of 10 letters).

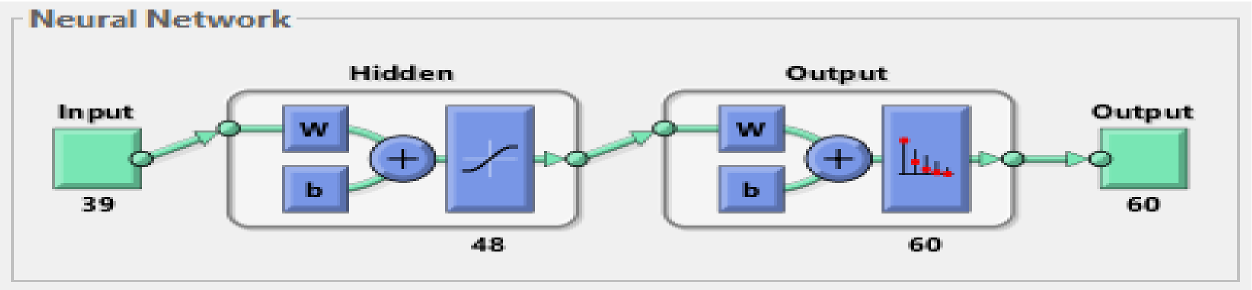


Figure (3.2): The designed neural network

The target data was selected here from the same MFCC features matrix but converted first into binary (logic) values (multiplied each element in matrix with -1 to eliminate the negative small decimal numbers for proper classification purpose) and then compared with the same train data input to the network.

Backpropagation neural networks was selected for this goal with one input layer (39 neurons equal to the number of cepstral coefficients), one hidden layer (48 elements) and one output layer (60 nodes for 60 classes of different vowels of 10 letters).

The following commands were witten to design the specification of neural networks and simulate it:

```
mynet=patternnet(48);
% Set up Division of Data for Training, Validation,
Testing
mynet.divideParam.trainRatio =100/100;
mynet.divideParam.valRatio = 0/100;
mynet.divideParam.testRatio = 0/100;
mynet.trainParam.epochs=1000;
mynet.trainParam.goal=1e-3;
mynet.trainParam.lr=0.01;
mynet.trainparam.min_grad=1e-25; %Diff. between epochs
net=train(mynet,xx,D1); % Train the Network
%Calculating the error (MSE)
error=sum(((vec2ind(net(xx)))-(vec2ind(D1))).^2)
a=net(xx) %To show the prob. values of the network
output
```

xx is the training data and D1 is the target data.

The total number learning epochs was 8000 times.

The error goal (Mean Square Error (MSE) was used) was to be `1e-25`.

The learning rate was 0.01.

There was a problem in accurately adjusting these training parameters, so getting low level of classification performance.

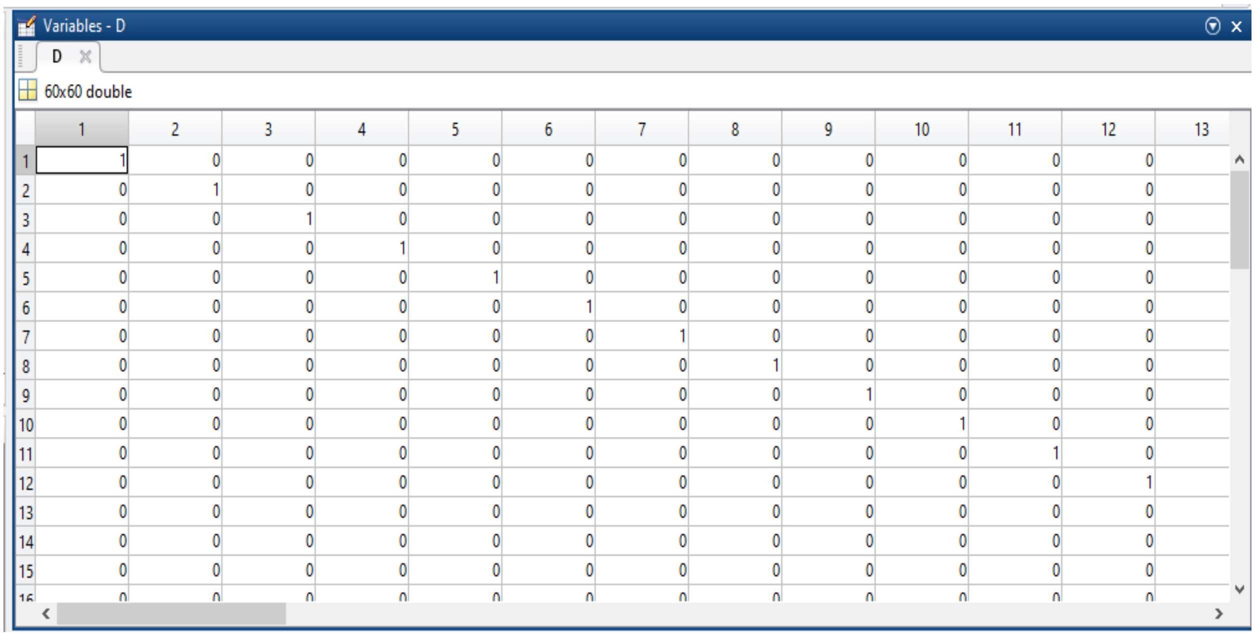
Therefore, the solution was in increasing up the number of input (spoken cepsrtal coefficients) but not increasing the number of hidden layers because of more trainng time delay might been occur.

The suggested number was 39 of cepstral coefficients and by default MATLAB considers about 60% of the input data for training, 20% for validation, i.e. to support the training and finally 20% for testing. The training percentage was taken to be 100% as first trial to achieve higher accuracy of recognition.

The goal was in increasing the size training data to ensure proper performance of fitting the goal line of classification (Y=T), where Y is the testing data and T is the target data.

The 39 cepstral coefficients of the 10 different spoken and recorded letters (6 vowels each), were applied to the input layer (39 neurons), trained, validated and then finally tested against the target cepstral coefficients gotten from the feature extraction process of the standard letters.

All training data frames examples has been resized to be of size (39 * 60) to unify them with the target data size.



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	0	0	0	0	0	0	0	0	0	0	0	
2	0	1	0	0	0	0	0	0	0	0	0	0	
3	0	0	1	0	0	0	0	0	0	0	0	0	
4	0	0	0	1	0	0	0	0	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	0	0	
6	0	0	0	0	0	1	0	0	0	0	0	0	
7	0	0	0	0	0	0	1	0	0	0	0	0	
8	0	0	0	0	0	0	0	1	0	0	0	0	
9	0	0	0	0	0	0	0	0	1	0	0	0	
10	0	0	0	0	0	0	0	0	0	1	0	0	
11	0	0	0	0	0	0	0	0	0	0	1	0	
12	0	0	0	0	0	0	0	0	0	0	0	1	
13	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	

Figure (3.3): Created target data for classification of ten letters database

Although there is no specific rule or concept to create the target data, i.e. just depending on the type of research problem or objective in other words, so here it was designed for the second experiment to be either 0 or 1, i.e. 1 to indicate the classified (target) letter and 0 to indicate nothing, i.e. not the target one.

For example, in figure (3.3), the first column indicate the target letter (Ain) (ع) with vowel Fatha, so the 1 in the 1st pixel is opposite immediately to the 1st output node and the rest of the rows in the 1st column are 0 to indicate that no other letter to be classified in this class.

3.4.3.3 Third experiment

In this experiment, the neural networks was used for training 66 examples of one letter e.g. Ain (ع) all loaded in one voice file (.wav).

The letter is spoken at first as 6 vowels, uttered one time with one letter, two letters, three letters, in complete word and finally in a sentence as shown in table (3.2).

Table (3.2): Database of Ain letter examples

No. of example	Description
1	عَ
2	عِ
3	عُ
4	عٌ
5	عًا
6	عٍـ
7	مَعَ
8	يَعْ
9	رَعْ
10	لَعَ
11	أَعِ
12	عَمِ
13	عَنْ

No. of example	Description
14	بَع
15	بِعَ
16	كَعُ
17	تَعْ
18	عَدْ
19	بَعَ
20	أَعُ
21	مَعُ
22	بَعْدِ
23	عَلَى
24	تَبِعَ
25	عَلِمَ
26	وَعَدَ
27	جَعَلَ
28	اَدْعُ
29	بَعْضَ
30	عِنْدَ
31	وُضِعَ
32	لَعَنَ
33	عِجْلَ
34	سَمِعَ
35	عَدُوّ
36	اَعْفُ
37	عِلمِ
38	عُمْي
39	عُفِيَ
40	بَيْع
41	عَذَاب
42	عَبْدِنَا

No. of example	Description
43	عَمِلُوا
44	عَرَضَهُمْ
45	عَفَوْنَا
46	يَدْعُونَ
47	لَعَلَّكُمْ
48	اَلْأَنْعَامِ
49	اَلْأَعْرَافِ
50	اَلرَّعْدُ
51	اَلْعَنْكَبُوتِ
52	اَلْمَعَارِجِ
53	فَاقِع
54	وَاسِع
55	اَلسَّمِيع
56	اَلْجُوع
57	اَلدَّاع
58	مَنَافِع
59	يَسْتَطِيع
60	اِسْتَطَاعَ
61	وَإِذَاَ اَلْقُبُورُ بُعْثِرَتْ
62	وَإِذَاَ اَلْعِشَارُ عُطِّلَتْ
63	وَاَللَّيْلِ إِذَا عَسْعَسَ
64	وَاَلْيَوْمِ اَلْمَوْعُودِ
65	إِنَّهُ عَلَى رَجْعِهِ لَقَادِر
66	وَإِذَاَ اَلْجَحِيمُ سُعِّرَتْ

It contains higher number of samples and frames (957 frames) and this is useful for achieving higher accuracy of classification.

The target data was designed as 2 * 957 size, i.e. two classes for each frame. Each 15 frames represents an example in file.

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 Introduction

This chapter explains the research practical results and discuss them.

4.2 Results Analysis and Discussion of the Implemented Voice Recognition System

In this section, the results of each experimental steps followed during the design and implementation of the Arabic voice letter recognition system were demonstrated here, analyzed and discussed.

Each of the following subtopics compares between the results of the standard (Ka’af) letter and the spoken (Ka’af) letter.

Only the letter (Ka’af) was used here for purpose of demonstration, so the same was applied for all the letters especially to the letters in database for this project.

4.3 Phase one (digital signal pre-processing):

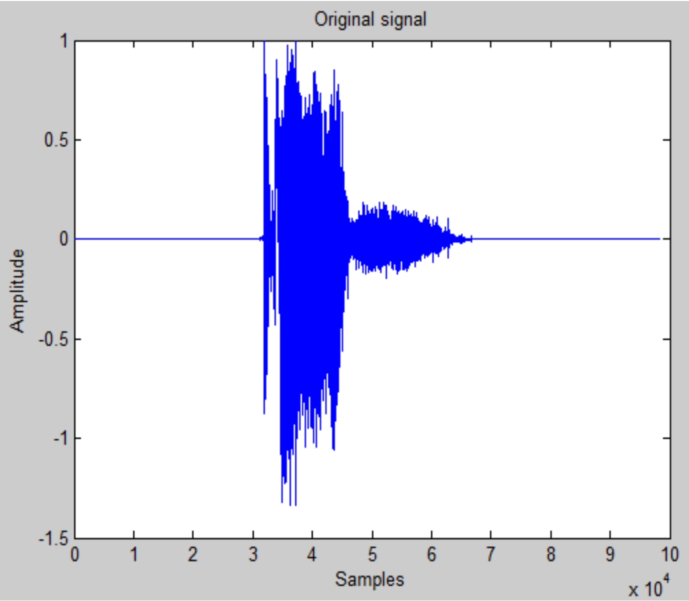


Figure (4.1): The standard (Ka’af) signal

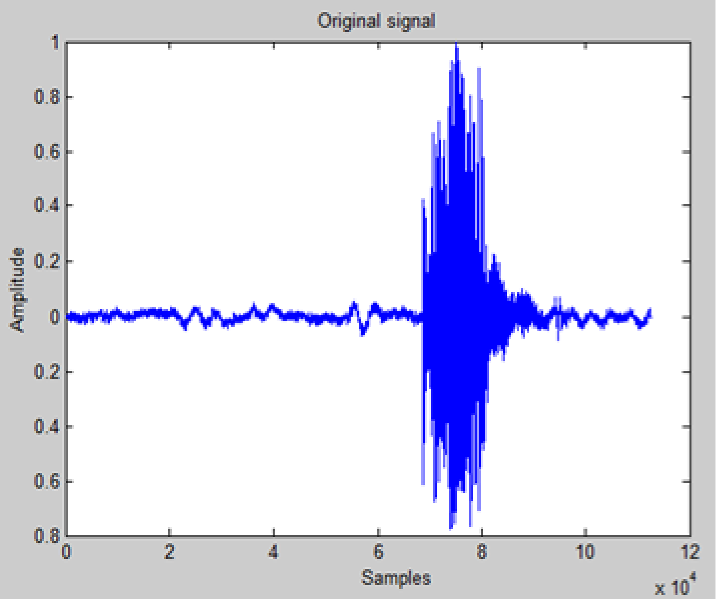


Figure (4.2): The spoken (Ka’af) signal

Both voice signals has 2 sec recorded clip. In figure (4.1), the signal contains about 100,000 samples while figure (4.2) has about 120,000 samples.

There was delay in recording the letter (ka’af) on the right signal, but both have similar shape and amplitude. The silence part on either sides differs between them slightly because having more noises on the right while very clear on the left because the voice was recorded using very high quality microphone and in advanced studios.

4.3.1 Step one (segmentation):

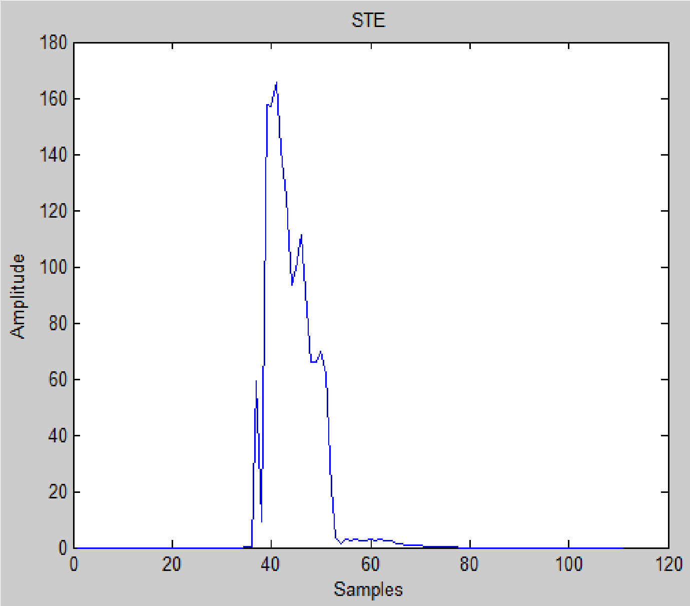


Figure (4.3): STE of standard (Ka'af) signal

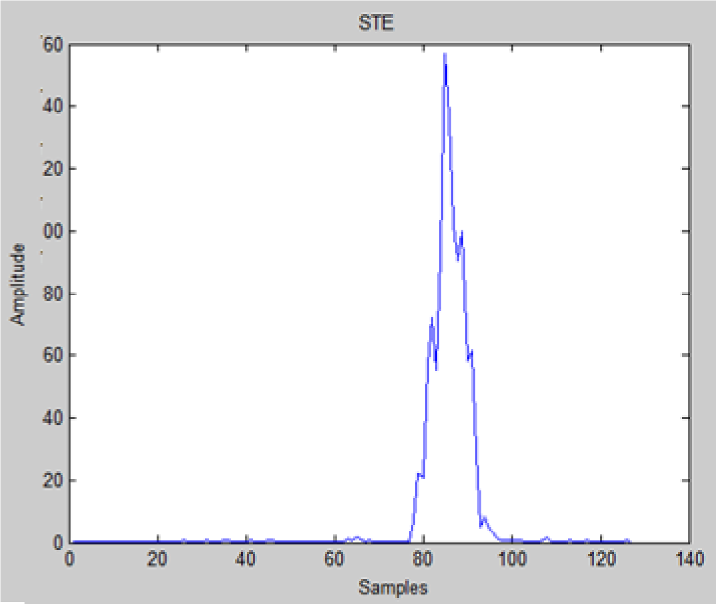


Figure (4.4): STE of spoken (Ka'af) signal

Both figures (4.3) and (4.4) shows the calculated STE for each frame in the signal. For proper scaling and silence removal, both STE's were normalized.

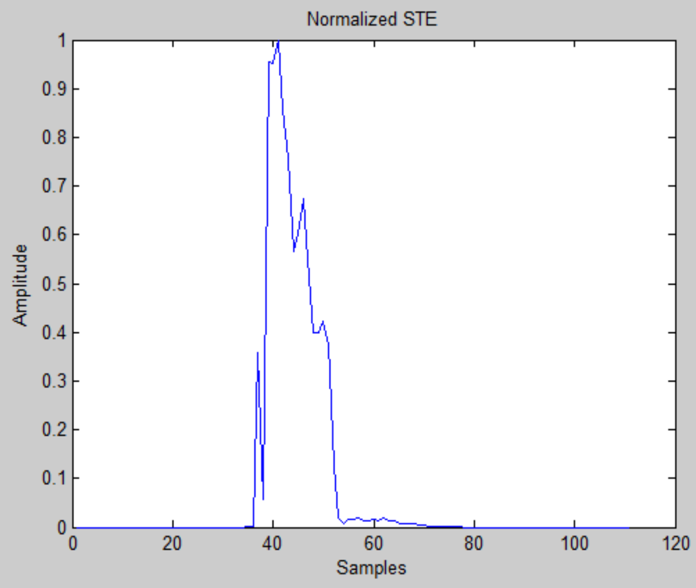


Figure (4.5): Normalized STE of standard (Ka'af)

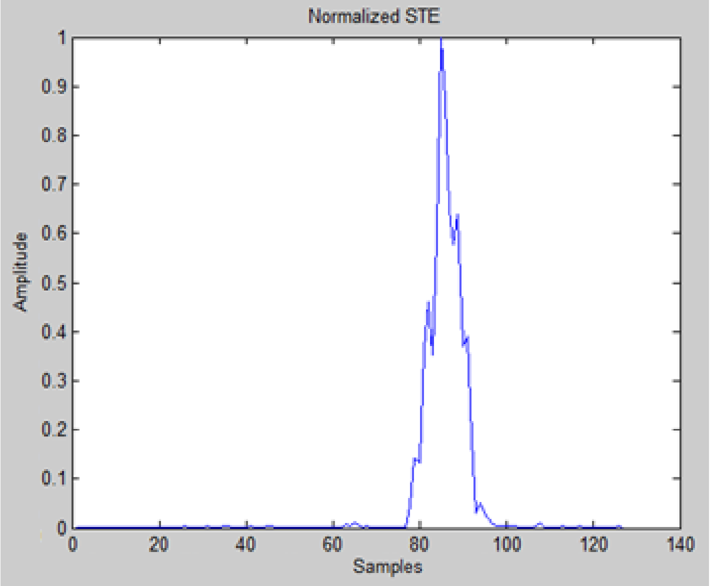
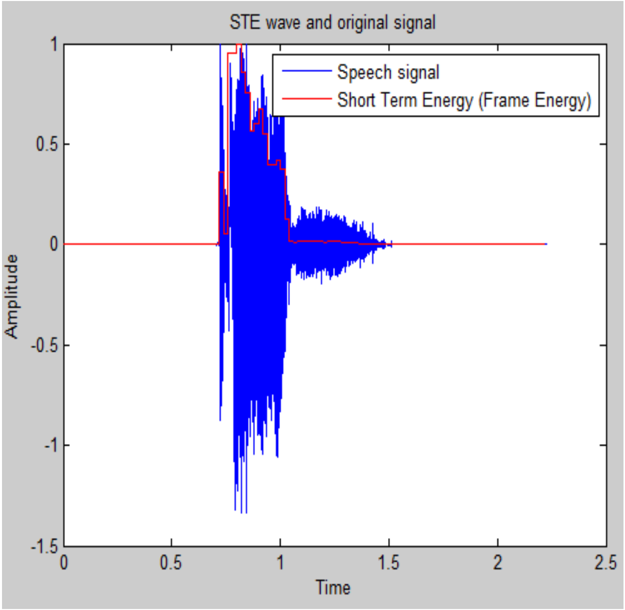


Figure (4.6): Normalized STE of spoken (Ka'af)



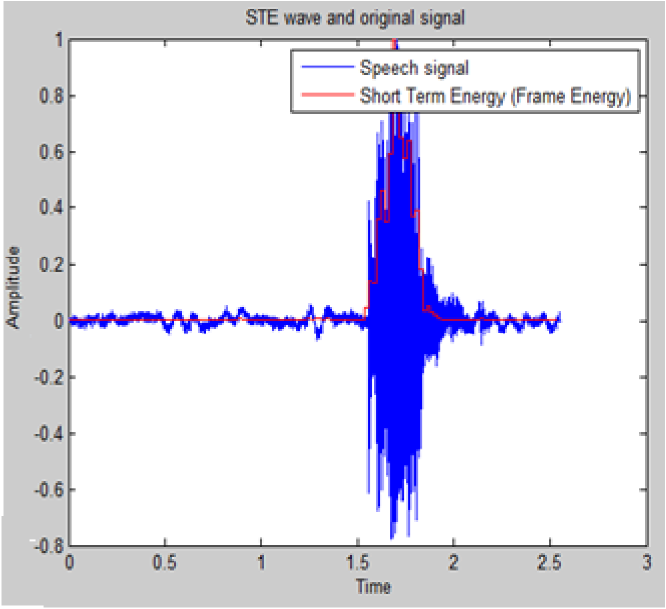
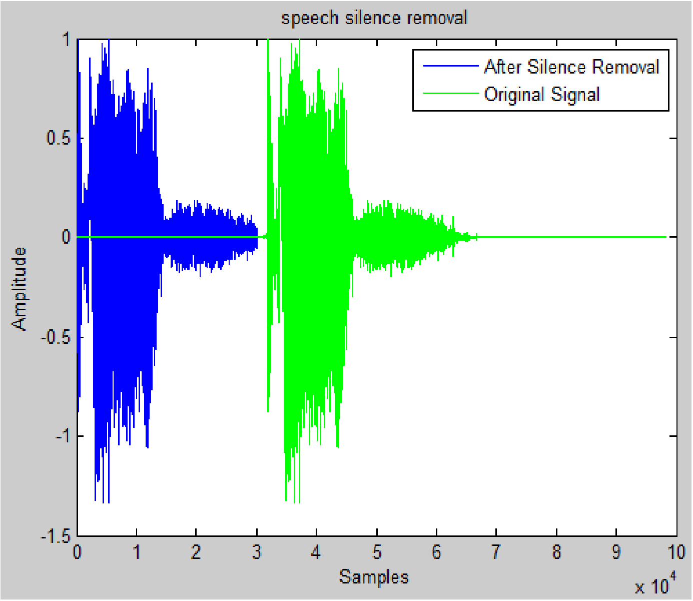


Figure (4.7): STE wave with standard (Ka'af)

Figure (4.8): STE wave with spoken (Ka'af)

The frame energy (STE) was displayed together as red step lines as shown in figures (4.7) and (4.8) to determine the threshold accurately of cutting the silence/noisy parts.



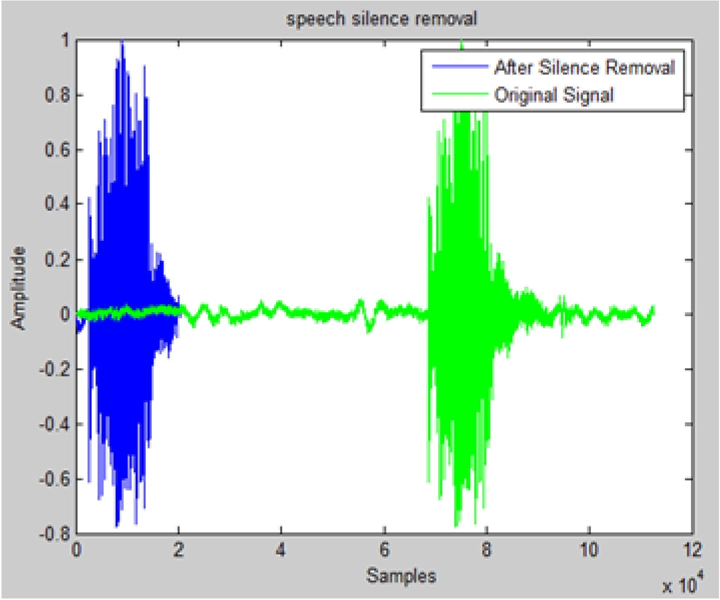
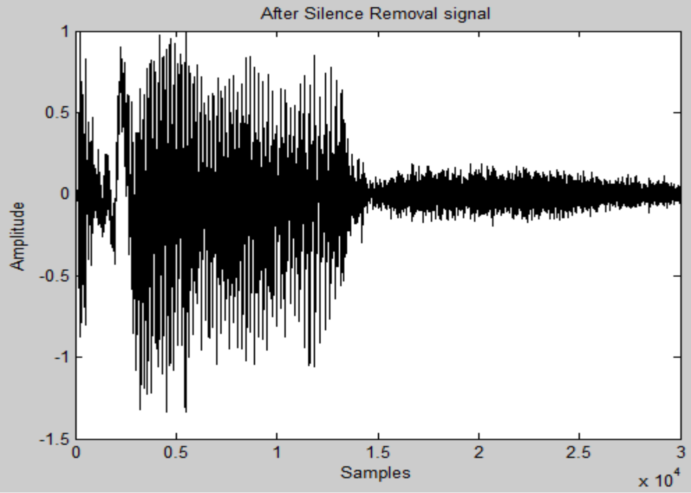


Figure (4.9): Standard (Ka'af) signal and signal after silence removal

Figure (4.10): Spoken (Ka'af) signal and signal after silence removal

The block signal on the right was cut for silence more than the left because of non-precise thresholding.



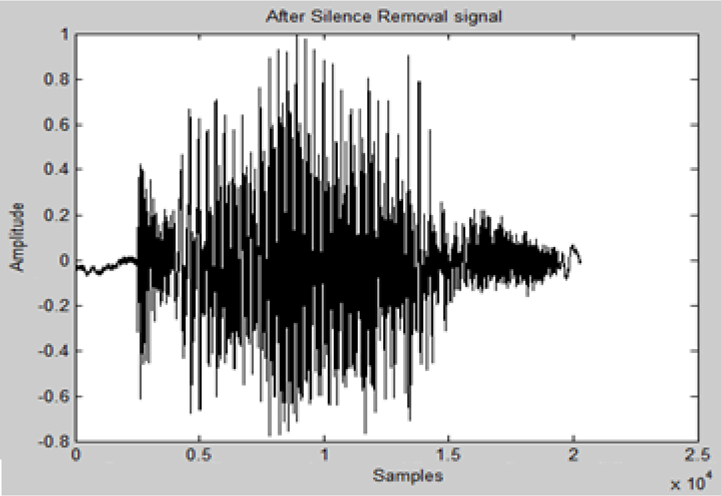
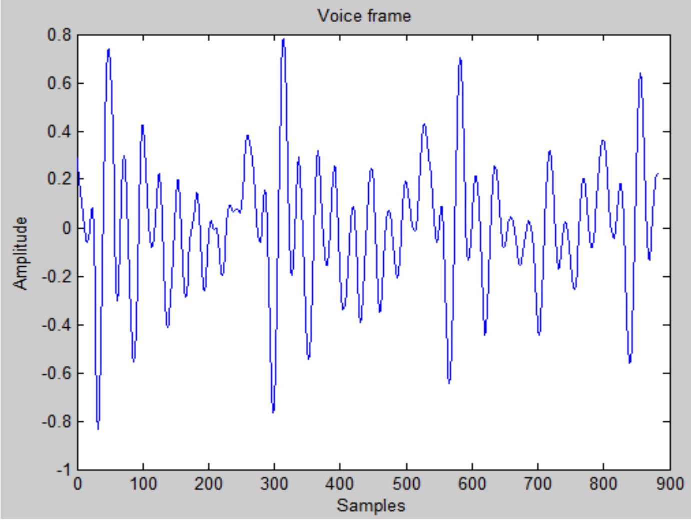


Figure (4.11): Standard (Ka'af) after silence removal *Figure (4.12): Spoken (Ka'af) after silence removal*

The results of silence removal from two signals of figures (4.11) and (4.12) were very efficient and both final signals were very similar to each other. Note that the total number of samples in both figures (4.1) and (4.2) was reduced to less than half as shown in figures (4.11) and (4.12).

4.3.2 Step two (digital signal pre-processing):

4.3.2.1 Frame blocking



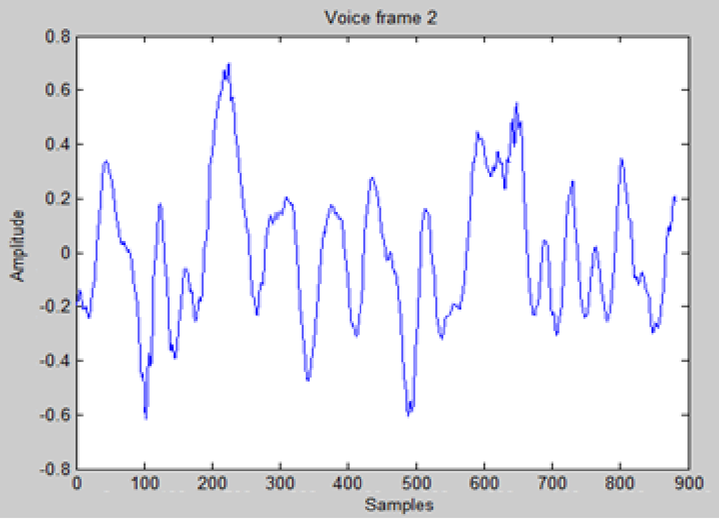


Figure (4.13): One frame of standard (Ka'af) *Figure (4.14): One frame of spoken (Ka'af)*

Both frames was selected to be at position of 15, the frame size of each has 882 samples and both look like periodic waveforms similar to each other.

4.3.2.2 Windowing and filtering

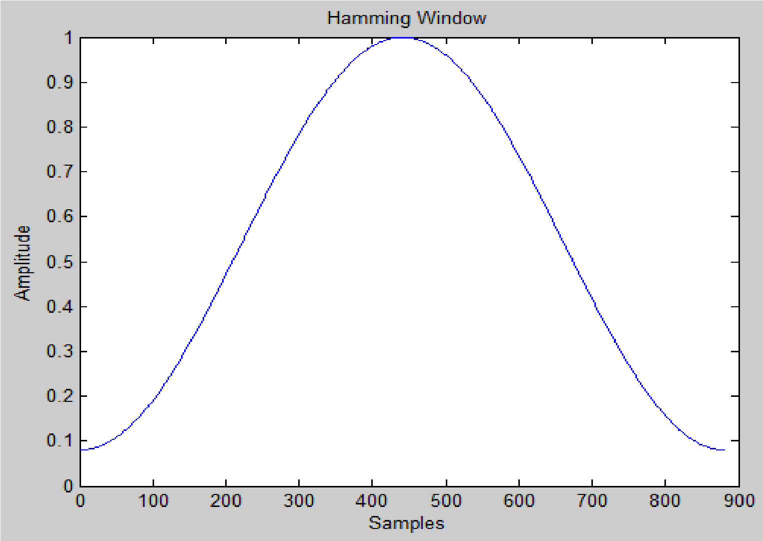


Figure (4.15): Hamming window function

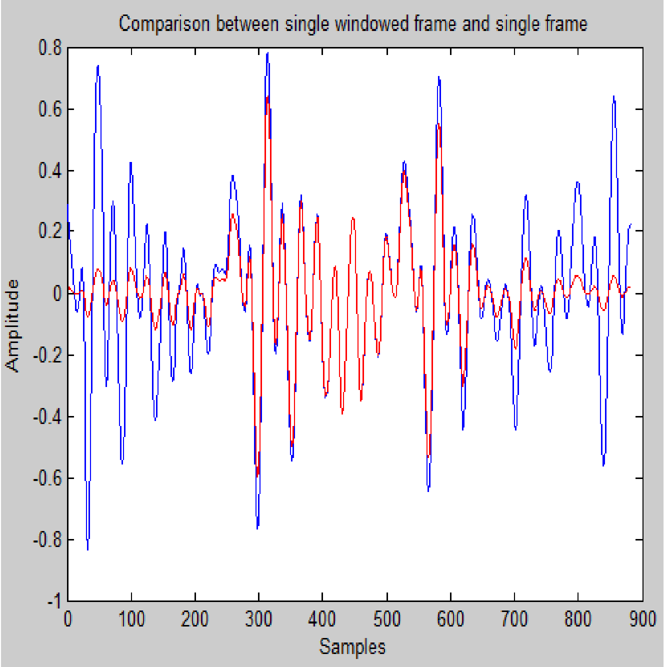


Figure (4.16): Windowed frame and frame of standard (ka'af)

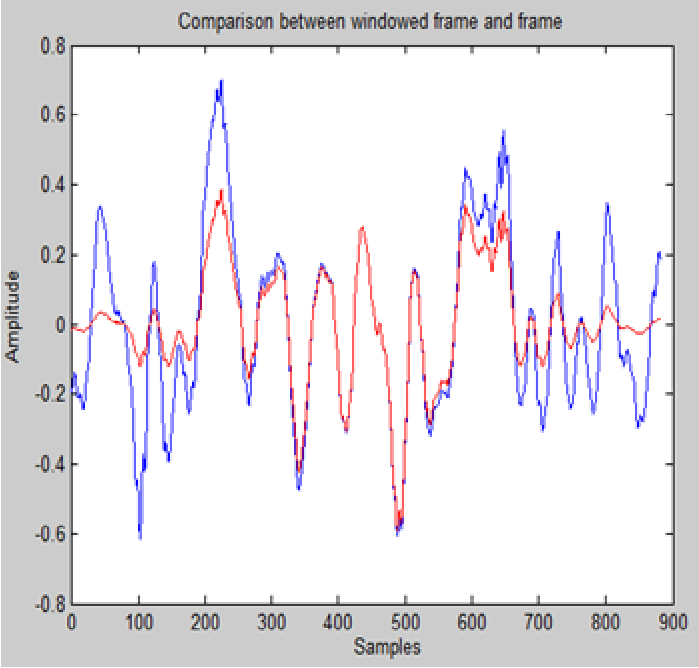


Figure (4.17): Windowed frame and frame of spoken (ka'af)

Both signals were reshaped properly after windowing in figures (4.16) and (4.17).

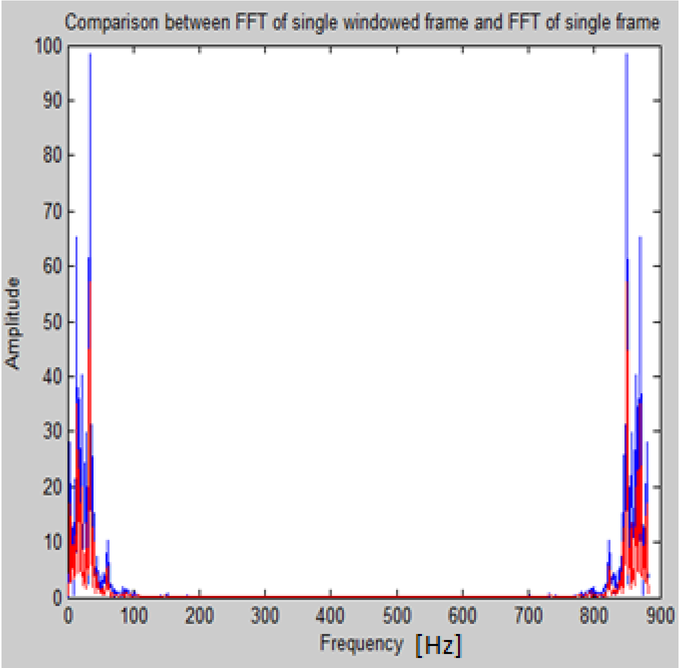


Figure (4.18): FFT of both windowed frame and frame of standard (ka'af)

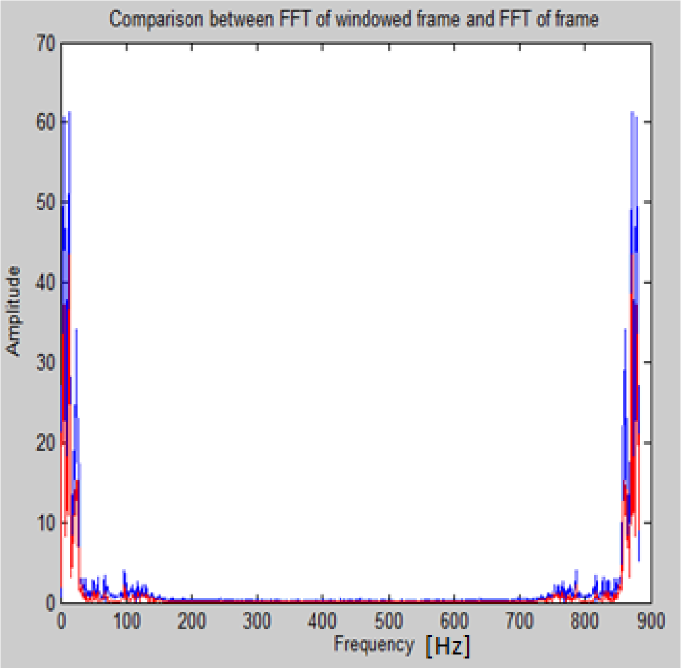


Figure (4.19): FFT of both windowed frame and frame of spoken (ka'af)

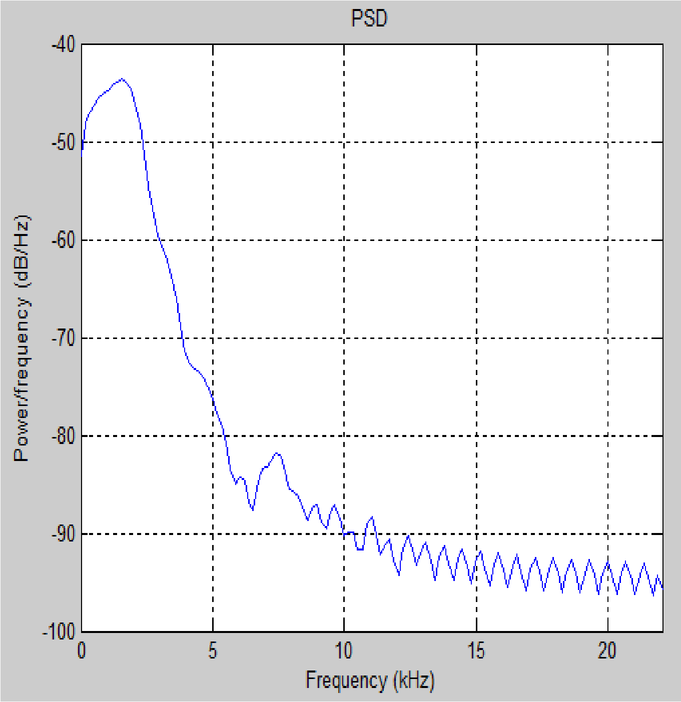


Figure (4.20): PSD of standard (Ka'af)

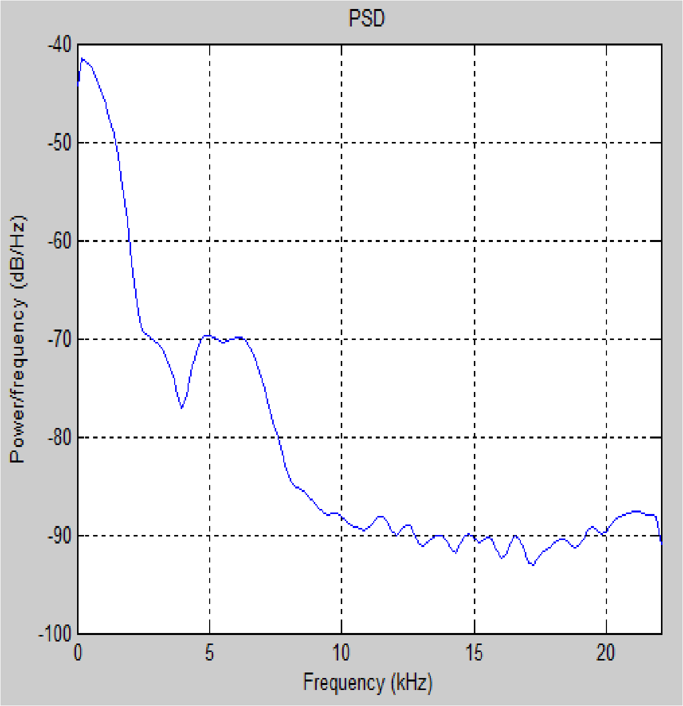


Figure (4.21): PSD of spoken (Ka'af)

In both figures (4.20) and (4.21), the energy of the voice signal damped gradually to lower values as the frequency increase.

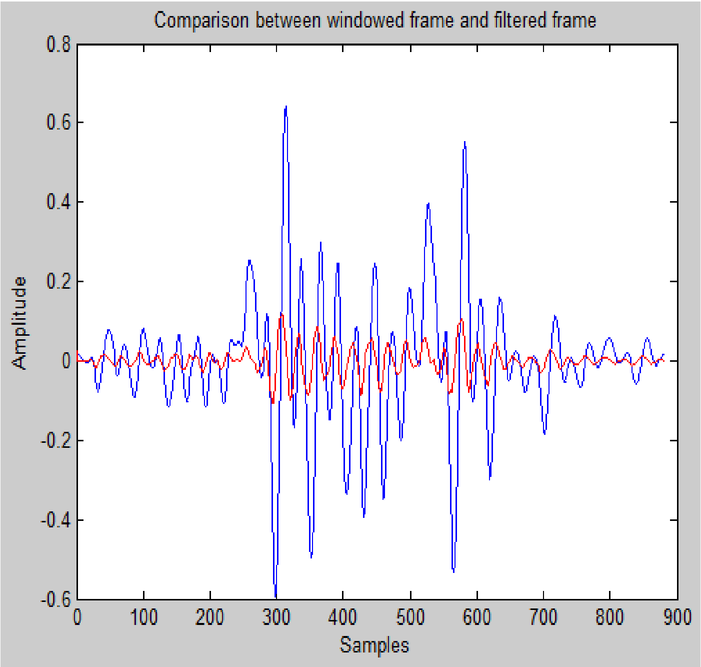


Figure (4.22): Windowed frame and filtered frame of standard (Ka'af)

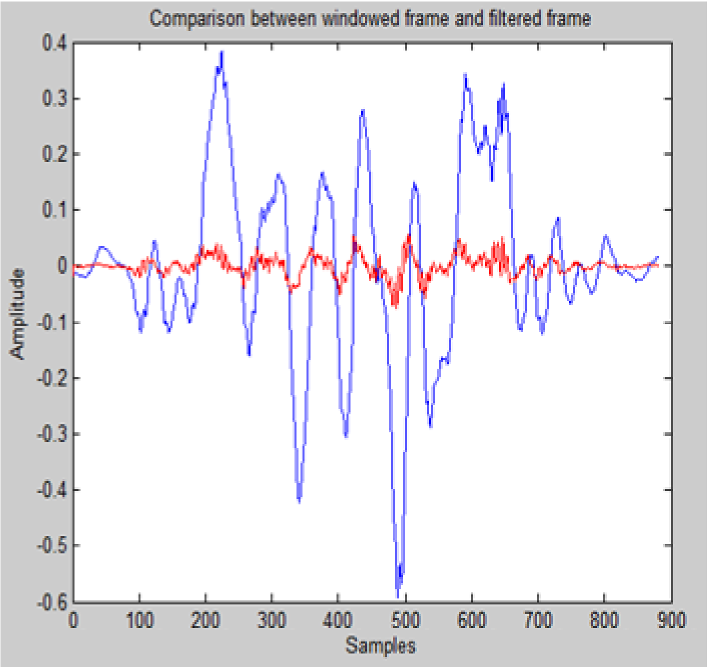


Figure (4.23): Windowed frame and filtered frame of spoken (Ka'af)

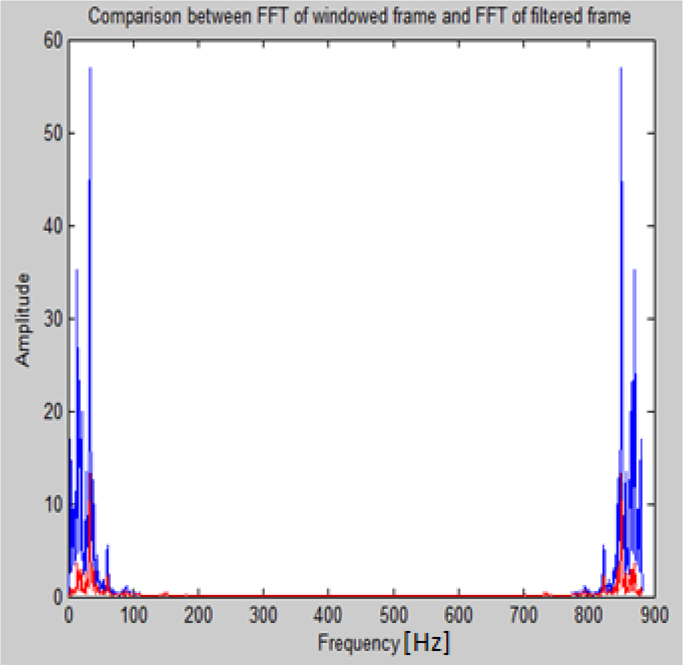


Figure (4.24): FFT of windowed frame and filtered frame of standard (ka'af)

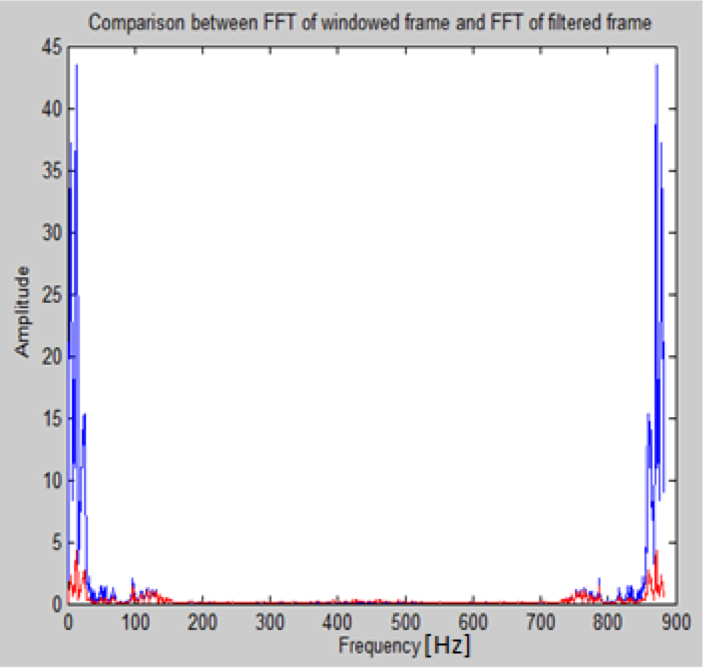


Figure (4.25): FFT of windowed frame and filtered frame of spoken (ka'af)

4.3.2.3 Fast Fourier Transform

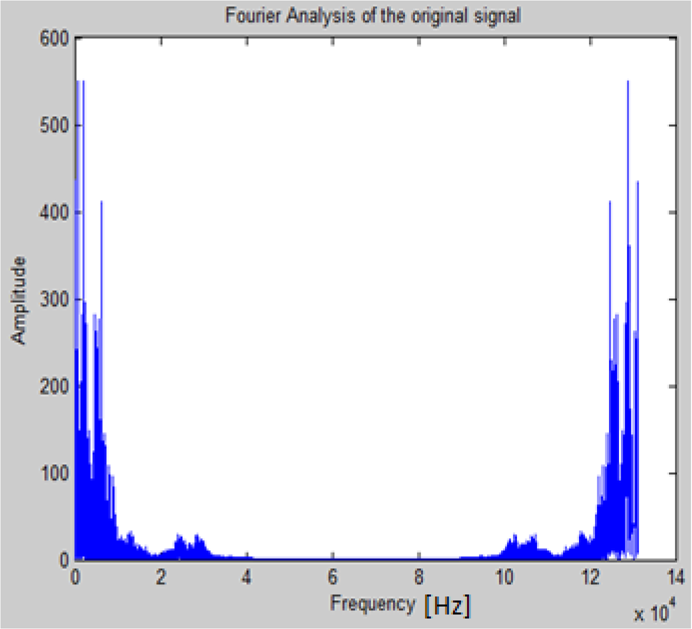


Figure (4.26): FFT of original standard (ka'af)

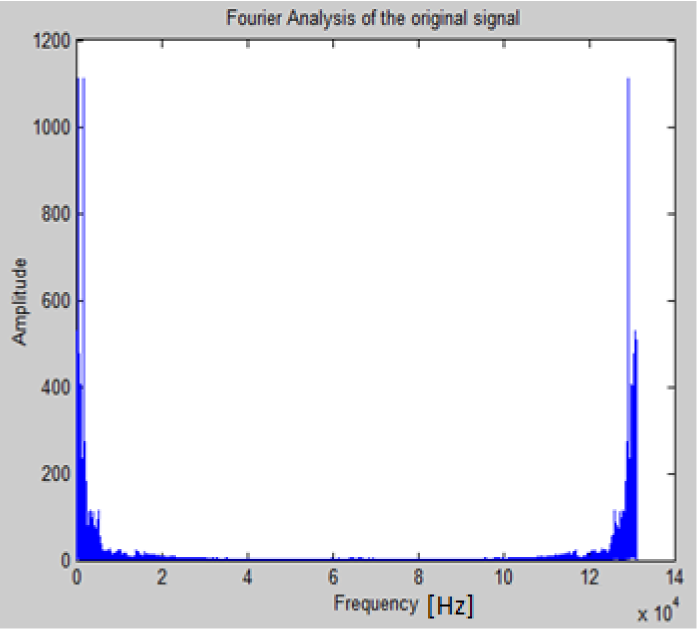


Figure (4.27): FFT of original spoken (ka'af)

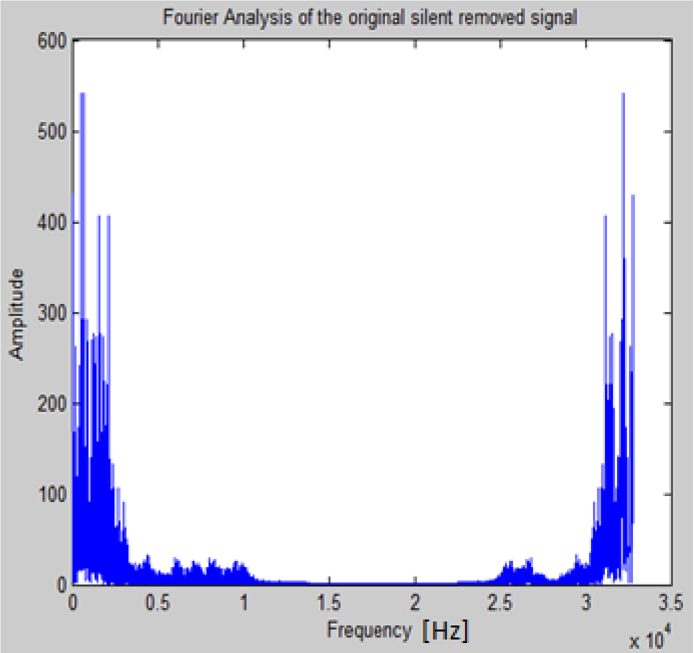


Figure (4.28): FFT of standard (ka'af)

(After silence removal)

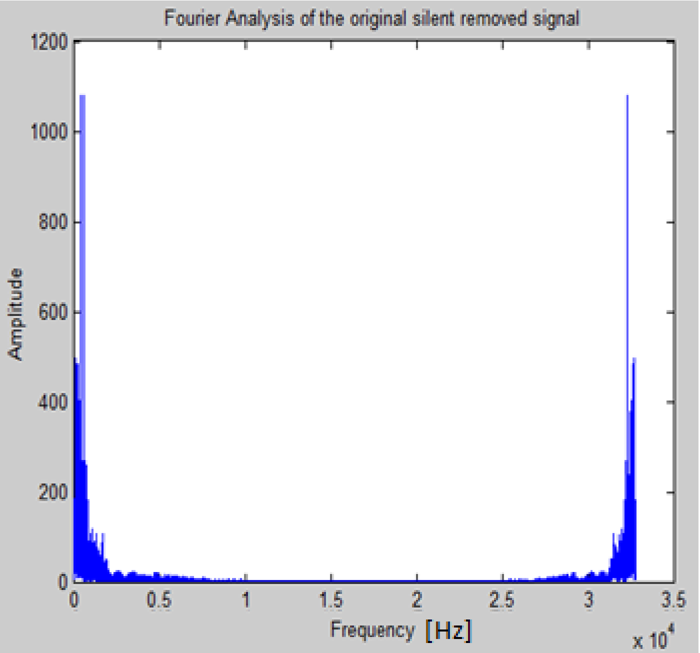


Figure (4.29): FFT of spoken (ka'af)

(After silence removal)

After silence removal, the high frequency band was increased as shown in figures (4.28) and (4.29) after FFT was applied.

4.4 Phase two (feature extraction):

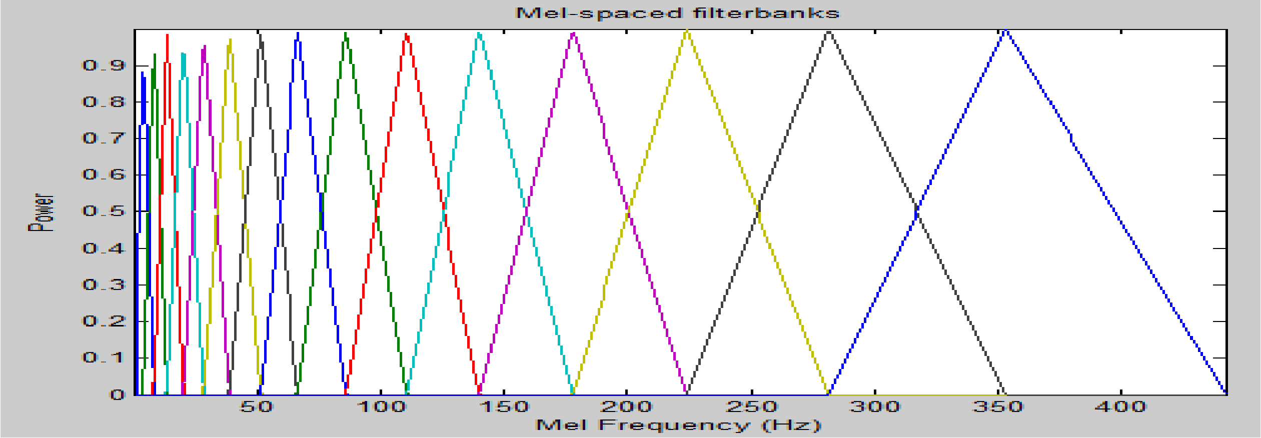


Figure (4.30): Mel spaced triangular filter banks of standard (ka'af)

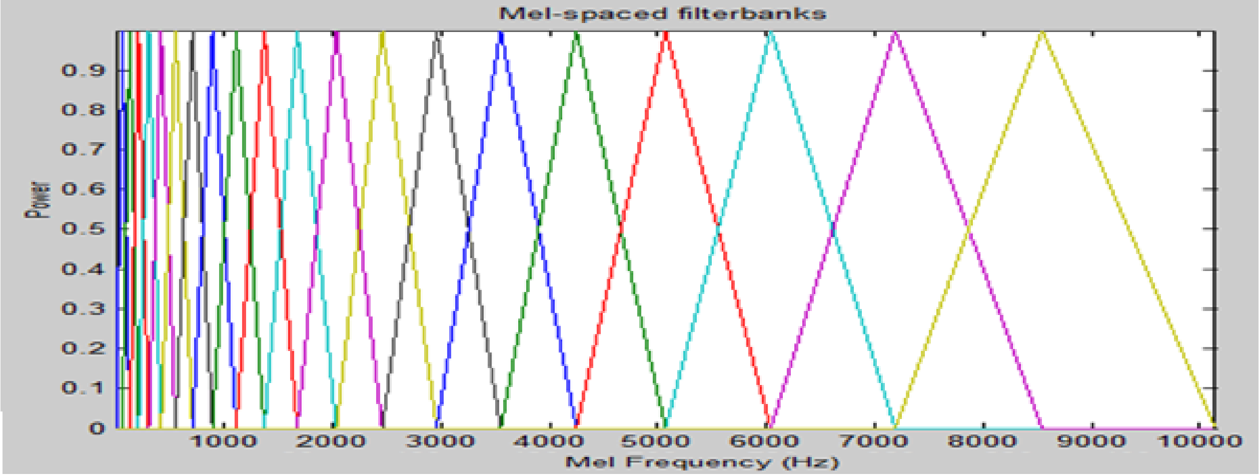


Figure (4.31): Mel spaced triangular filter banks of spoken (ka'af)

In figure (4.31), the filter banks was increased in their bands as they move to the right which means more useful information at high frequency elements.

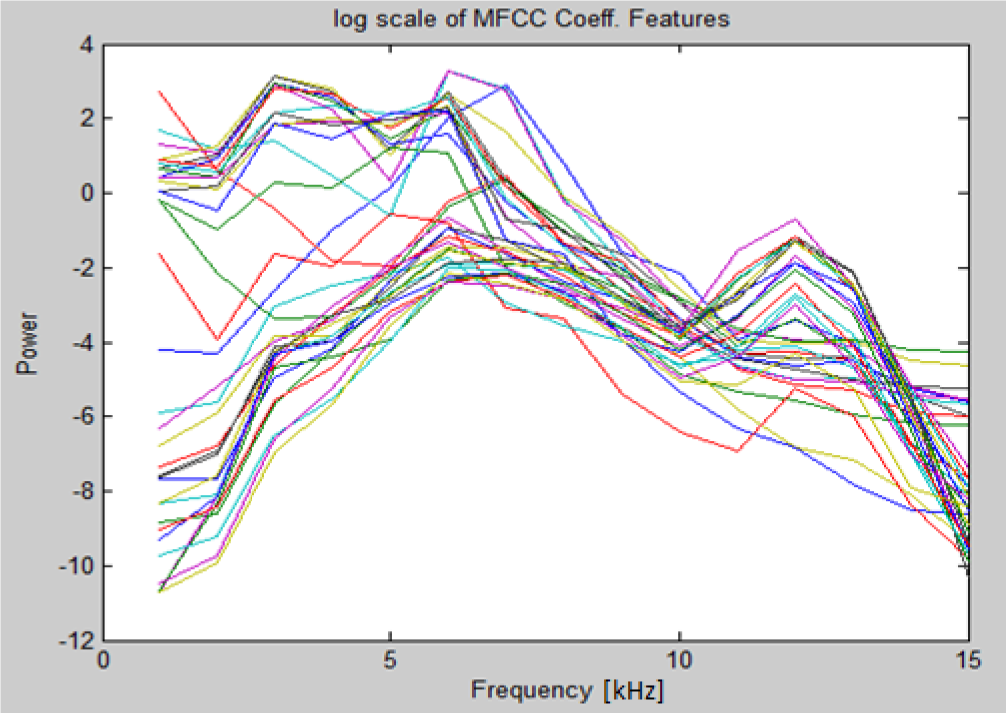


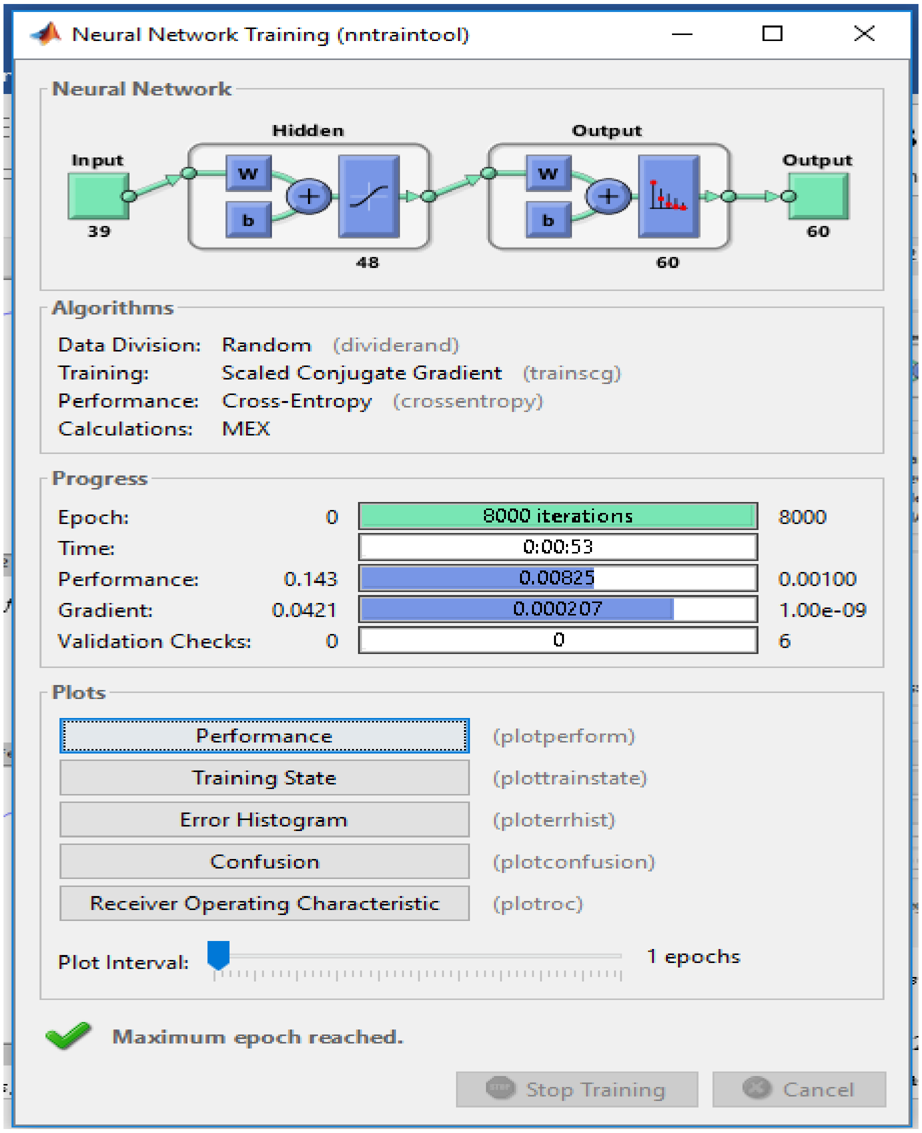
Figure (4.32): Log scale of MFCC coefficients features of standard (ka'af)

The above figure (4.32) demonstrates the log power scale of MFCC confidents in each filter bank for all the frames created in the signal of recorded voice of “Ka’af” and the same thing can be applied to the rest of all letters. After the frequency 30 Hz, most of the curves tend to be at steady state between the range (-5 dB – (-10) dB) but before it there was fluctuations in dB power around 0 dB and (-5) dB. This figure (4.32) showed the inverse relationship between the frequency banks and power, i.e. with increasing the frequency the power in dB will decreased gradually till reaching the most possible lower values, i.e. (-10 dB) and then tend to stabilize.

4.5 Phase three (classification):

Since the same techniques have conducted for all the experiments, the results of experiment two have only be demonstrated here in details for just purpose of summarizing.

The network was trained 8000 times of epochs with a goal reached 1e-9. The actual goal of performance was achieved 0.0089 at epoch 1000. The network needs to be trained with more times of iterations to achieve the maximum degree of accuracy.



Algorithms

Data Division: Random (dividerand)
Training: Scaled Conjugate Gradient (trainscg)
Performance: Cross-Entropy (crossentropy)
Calculations: MEX

Progress

Epoch:	0	8000 iterations	8000
Time:		0:00:53	
Performance:	0.143	0.00825	0.00100
Gradient:	0.0421	0.000207	1.00e-09
Validation Checks:	0	0	6

(plotperform)
(plottrainstate)
(ploterrhist)
(plotconfusion)
(plotroc)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Figure (4.33): GUI of neural networks for training 39 MFCC coefficients of each frame of each letter in the database.

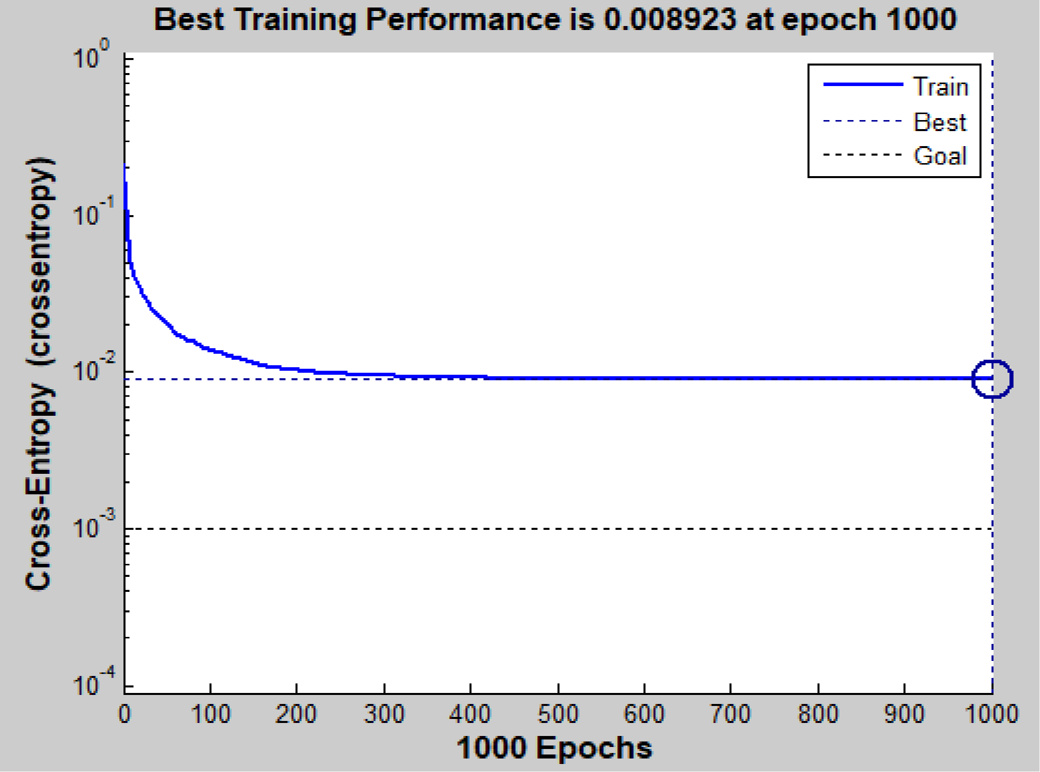


Figure (4.34): Best training performance plot of MFCC coeff.

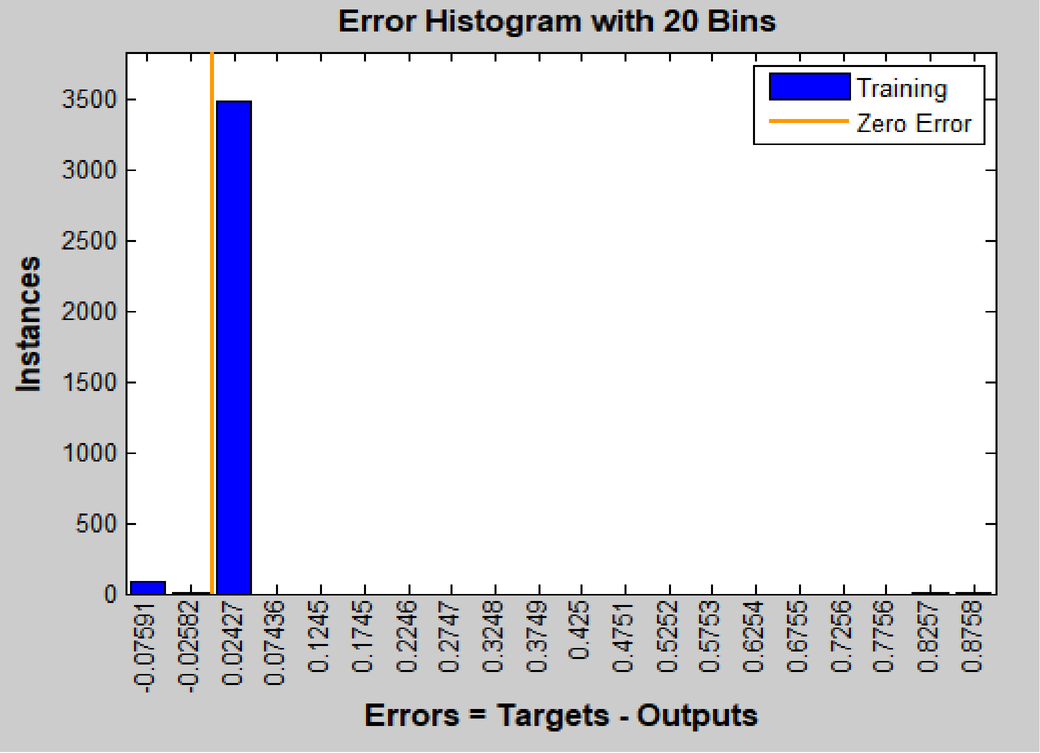


Figure (4.35): Error histogram of trained neural networks

The maximum accuracy reached in this project was 87% which is semi-perfect results because the neural network needs to be adjusted and designed more properly to learn perfectly.

In figure (4.35), the variation of errors was distributed semi-normally which stands for enhanced design and proper MFCC features extraction was obtained from this system.

Table (4.1): Comparison of both overall accuracy and error between the three experiments

No. of experiment	Overall accuracy achieved	Error
Experiment one	65%	35%
Experiment two	87%	13%
Experiment three	95%	5%

From table (4.1), the highest overall accuracy was achieved for experiment three because it has higher amount of input data examples although the semi-adjustment design procedures with other experiment one and two but the last both have been designed for less number of input data.

The experiment one was the lowest accuracy because it has only conducted for one consonant letter, so it was difficult to recognize it properly.

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter contains the final achieved conclusion of the work done in this research and mention some of the recommended hints for future work.

5.2 Conclusion

Voice letter recognition system designed consists of three different phases: Digital signal processing, voice features extraction and machine learning algorithm for classification.

Mel frequency filter banks of 39 non-linearly spaced, triangular band pass filters with overlapping bandwidths; when applied to the spectral magnitudes of FFT yields the dominant frequency components (or peaks or formants) for each frame of the input speech signal. MFCC algorithm performed on ‘wav’ files in MATLAB yields a matrix with number of columns equal to number of frames, which is determined by the size of input file and number of row equal to the DCT size, which is 39 in our case for voice signal.

The back propagation multilayer neural networks have designed and tested among three different experiments and trained much of times to approach higher degree of accuracy.

The overall accuracy obtained from this classification was 65% with an error of 35% for one consonant letter, 87% accuracy and an error of 13% for 10 isolated different letters and 6 vowels each and finally 95% accuracy and an error of 5% for 66 different examples of one letter (vowels, words and sentences) stored in one voice file.

5.3 Recommendations

Below are some recommendations for future work:

1- Develop hybrid classifier of neural networks and Hidden Markov Model (HMM) to enhance the accuracy of phenome recognition.
2- Try coding with Python language since it is the most applicable and simple programming language for machine learning.
3- Develop this system for learning Tajweed and Tellawah of Qur’a’n.
4- Design an attractive Graphical User Interface (GUI).

REFERENCES

[1] Moaz Abdulfattah Ahmad, Rasheed M. El Awady, *“Phonetic Recognition of Arabic Alphabet letters using Neural Networks”,* International Journal of Electric & Computer Sciences IJECS-IJENS Vol: 11 No: 01, 2011.

[2] Elvira Sukma Wahyuni, *“Arabic Speech Recognition Using MFCC Feature Extraction and ANN Classification”,* 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017.

[3] Ayat Hafzalla Ahmed, Sherif Mahdi Abdo, *“ Verification System for Quran Recitation Recordings”,* International Journal of Computer Applications (0975 – 8887) Volume 163 – No 4, April 2017.

[4] Safaa Omer Mohammed Nssr, *“Voice Recognition by using Machine Learning A Case Study of some Rules of Tajweed”,* Sudan University of Science and Technology College of Graduate Studies, 2016.

[5] Øystein Staven, *“Detection of phonetic features for automatic classification of Norwegian Dialects”,* Norwegian University of Science and Technology, Master of Science in Electronics, June 2016.

[6] Tayseer Mohammed Hasan Asda, Teddy Surya Gunawan, Mira Kartiwi, Hasmah Mansor, “*Development of Quran Reciter Identification System Using MFCC and Neural Network*”, TELKOMNIKA Indonesian Journal of Electrical Engineering, Jan 2016.

[7] Kavita Yadavi, Moresh Mukhedkar, “*MFCC Based Speaker Recognition using Matlab”,* MFCC Based Speaker Recognition using Matlab, 2014.

[8] Hassan M. H. Mustafa, Fadhel Ben Tourkia, "*On Comparative Study for Two Diversified Educational Methodologies Associated with "How to Teach Children Reading Arabic Language?" (Neural Networks' Approach*)", Open Access Library Journal, 2016.

[9] Yakubu A. Ibrahim, Tunji S. Ibiyem, "*A Study on Efficient Automatic Speech Recognition System Techniques and Algorithms*", Anale. Seria Informatică. Vol. XVI fasc. 2, 2018.

[10] Nilu Singh, R. A. Khan, *"Digital Signal Processing for Speech Signals",* Digital Signal Processing for Speech Signals, 2015.

[11] Siddhant C. Joshi, Dr. A.N.Cheeran, "MATLAB Based Feature Extraction Using Mel Frequency Cepstrum Coefficients for Automatic Speech Recognition", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 6, June 2014

[12] Moore. Holly, *'MATLAB for engineers',* P:1-2, 3rd ed., Pearson Education, Inc., publishing as Prentice Hall, 2012.

[13] Safdar Tanweer, Abdul Mobin, Afshar Alam, *'Analysis of Combined Use of NN and MFCC for Speech Recognition',* International Journal of Computer and Information Engineering, Vol: 8, No:9, 2014

[14] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, 'Spoken Language Processing', ISBN 0-13-022616-5, publishing as Prentice Hall, 2001.

APPENDIX A

MATLAB code of spoken letter recognition system

```
function output = Myproject(~)
%% Load the audio file
[fname path]=uigetfile('*.*','Enter voice');  %To find
the matlab path
fname=strcat(path,fname);
%[voice,fs]=audioread('22-kaaf.mp3');
[voice,fs]=audioread(fname); %To read the audio file
%gn = sum(g, 2) / size(g, 2); %convert to mono
sound(voice,fs) %To play the voice
voice=voice(:,1)./abs(max(voice(:,1))); %Normalize data
figure(1);
plot(voice)
title('Original signal')

%% Fourier of the original signal
l=length(voice);
nfft=2^nextpow2(l);
f=fs/2*linspace(0,1,nfft/2+1);
xf=abs(fft(voice,nfft));
figure(2);
plot(xf);
hold on;
plot(f,xf(1:nfft/2+1));
title ('Fourier Analysis of the original signal')

%% Do framing
fd1=0.02; %frame duration in seconds (20 msec)
fs=44100; %sampling frequency
f_size1=floor(fd1*fs); % No. of samples in the frame
n1=length(voice); % No. of samples in the original
signal
n_f1=floor(n1/f_size1); %Total No. of frames in the
original signal

% Total no. of frames = Total number of samples in the
signal (n) / frame size

temp1=0;
```

```
for i=1:n_f1
    frames(i,:)=voice(temp1+1:temp1+f_size1);
    temp1=temp1+f_size1; %counter
end
figure(3)
plot(frames(1,:)); % choose frame number 1
title('Zero frame 1')
figure(4)
plot(frames(50,:)); % choose frame number 50
title('Voice frame 1')

%% STE Method for silence removal
[r,~]=size(frames);
for i=1:r  %ste for specific frame
    ste(i)=sum(frames(i,:).^2);
end
figure(5)
plot(ste);
title('STE')
ste=ste./max(ste); %Normalize ste
figure(6)
plot(ste);
title('Normalized STE')
ste_wave=0;
for j=1:length(ste) %ste for all frames
    b=length(ste_wave);
    ste_wave(b:b+f_size1)=ste(j);
end
t=0:1/fs:length(voice)/fs; %plot ste with the original
signal
t=t(1:end-1);
t1=0:1/fs:length(ste_wave)/fs;
t1=t1(1:end-1);
figure(7)
plot(t,voice');
hold on;
plot(t1,ste_wave,'r')
title('STE wave and original signal')
legend('Speech signal','Short Term Energy (Frame
Energy)');
```

```
%% Silence Removal
id=find(ste>=0.002);
fr_ws=frames(id,:);
re=reshape(fr_ws',1,[]);
figure(8)
plot(re); title('speech silence removal');
hold on;
plot(voice,'g');
legend('After Silence Removal','Original Signal');
figure (9);
plot(re,'k'); title('After Silence Removal signal');

%% Do reframing 2
fd2=0.02; %frame duration in seconds (20 msec) selected
from 2 seconds
f_size2=floor(fd2*fs); % No. of samples in the frame
n2=length(re); % No. of samples in the silent removed
signal
n_f2=floor(n2/f_size2); %Total No. of frames in the
silent removed signal
temp2=0; %temporary value
for i=1:n_f2
    fr(i,:)=re(temp2+1:temp2+f_size2);
    temp2=temp2+f_size2; %counter
end
figure(10)
plot(fr(10,:)); % plot frame number 15 as example
title('Voice frame 2')

%% Fourier of the silent removed signal
l=length(re);
nfft=2^nextpow2(l);
f=fs/2*linspace(0,1,nfft/2+1);
xf=abs(fft(re,nfft));
figure(11);
plot(xf);
hold on;
plot(f,xf(1:nfft/2+1));
title ('Fourier Analysis of the original silent removed
signal')
```

```
%% windowing (Frames are windowed to improve the
frequency domain representation)
figure(12);
plot(hamming(length(frames(50,:))))
title('Hamming Window')
fr_win=frames(50,:)'.*hamming(length(frames(50,:)));
figure(13);
plot(frames(50,:)); hold on; plot(fr_win,'r')
title('Comparison between single windowed frame and
single frame')
figure(14);
plot(abs(fft(frames(50,:)))); hold on;
plot(abs(fft(fr_win)),'r')
title('Comparison between FFT of single windowed frame
and FFT of single frame')

%% PSD (Power Spectrum Density)
h=spectrum.welch;
d=psd(h,fr_win,'Fs',fs);
figure(16);
plot(d)
title('PSD')

%% Spectrogram of the windowed and silence removed
signal
spectrogram(fr_win,[],[],[],fs,'yaxis'); colorbar;
%plot the power levels in our signal
%shading flat,

%% Preemphasis
alpha=[1 -0.95]; %preemphasis (HPF) coefficient
figure(17);
bode(alpha,1);
y=filter(alpha,1,fr_win);
figure(18);
plot(fr_win);
hold on;
plot(y,'r')
title('Comparison between windowed frame and filtered
frame')
figure(19);
```

```
plot(abs(fft(fr_win)));
hold on;
plot(abs(fft(y)),'r')
title('Comparison between FFT of windowed frame and FFT
of filtered frame')

%% Feature extraction using MFCC

MF = @(f) 2595.*log10(1 + f./700); %The mel scale
filter bank formula
invMF = @(m) 700.*(10.^(m/2595)-1);

M=39;
fbegin=0;

M = M+2; % number of triangular filters
mm = linspace(MF(fbegin),MF(fs/2),M); % equal space in
mel-frequency
ff = invMF(mm); % convert mel-frequencies into
frequency

X = abs(fft(fr'));%fft of filtered frames
N = length(X); % length of a short time window
N2 = max([floor(N+1)/2 floor(N/2)+1]); %Calculate
energy excluding the negative frequency part (the
second half), therefore I take only frequencies between
0-4kHz
P = 1/N*abs(X(1:N2,:)).^2; % NoFr no. of periodograms
(Normalized power)
mfccShapes = triangularFilterShape(ff,N,fs); %Get a
bank of 10 triangularly shaped filters with centres
spread over mel frequency between 20Hz and 4kHz

%output=mfccShapes;    %Get mfcc matrix
figure(20);
plot(mfccShapes)    % Plot mfcc matrix
title ('Mel-spaced filterbanks')
axis tight
```

```
output= log(mfccShapes'*P); %Apply filters on the
vector of energy, sum the vectors and take natural
logarithm
figure(21);
plot(output)
title ('log scale of MFCC Coeff. Features')

function [out,k] = triangularFilterShape(f,N,fs)
    N2 = max([floor(N+1)/2 floor(N/2)+1]);
    M = length(f);
    k = linspace(0,fs/2,N2);
    out = zeros(N2,M-2);
    for m=2:M-1
       I = k >= f(m-1) & k <= f(m);
        J = k >= f(m) & k <= f(m+1);
        out(I,m-1) = (k(I) - f(m-1))./(f(m) - f(m-1));
        out(J,m-1) = (f(m+1) - k(J))./(f(m+1) - f(m));
    end
end
save('Projects-variables')
end
```

APPENDIX B

MATLAB code for Neural networks design (M-file one)

```
function [net,error] = Neural2(T, D)
%% creating training and targeting data
N2 = 60;
for K = 1:N2
    x=imresize(T(:,:,K),[39 60]);
    %xx=reshape(T(:, :, K),[],1);
    Target_data = D(K, :)';
    % Neural Networks Design
    mynet=patternnet(78);
    % Set up Division of Data for Training, Validation, 
Testing
    mynet.divideParam.trainRatio =100/100;
    mynet.divideParam.valRatio = 0/100;
    mynet.divideParam.testRatio = 0/100;
    mynet.trainParam.epochs=1000;
    mynet.trainParam.goal=1e-3;
    mynet.trainParam.lr=0.01;
    mynet.trainparam.min_grad=1e-9; %Diff. between 
epochs
    % Train the Network
    net=train(mynet,x,Target_data);
    %Calculating the error (MSE)
    error=sum(((vec2ind(net(x)))-
(vec2ind(Target_data))).^2)
    %simulate the output of the network (Compare 
between the actual and desired outputs)
    %Y=sim(net,Target_data);
    %figure (1);
    %plot(TRANS_EST,EMIS_EST,'k'); hold on; 
plot(TRANS_EST,Y,'r')
    a=net(x);

    %Note: Target, T, is the desired output for the 
given input, X.
    %Train the network with known input (X) and target 
(T).
```

```
    %The output of the resulting design, given the
input, is output , Y.
    %The error is e = T-Y. Of course the most common
ultimate goal of training is to minimize the mean-
squared-error.
end
plotconfusion(Target_data,net)
save('Neural-variables','net')
end
```

APPENDIX C

MATLAB code for Neural networks design (M-file two)

```
N2 = 60;
for K1 = 1:N2
    T1=imresize(T1,[39 60]);
    T2=imresize(T2,[39 60]);
    T3=imresize(T3,[39 60]);
    T4=imresize(T4,[39 60]);
    T5=imresize(T5,[39 60]);
    T6=imresize(T6,[39 60]);
    T7=imresize(T7,[39 60]);
    T8=imresize(T8,[39 60]);
    T9=imresize(T9,[39 60]);
    T10=imresize(T10,[39 60]);
    T11=imresize(T11,[39 60]);
    T12=imresize(T12,[39 60]);
    T13=imresize(T13,[39 60]);
    T14=imresize(T14,[39 60]);
    T15=imresize(T15,[39 60]);
    T16=imresize(T16,[39 60]);
    T17=imresize(T17,[39 60]);
    T18=imresize(T18,[39 60]);
    T19=imresize(T19,[39 60]);
    T20=imresize(T20,[39 60]);
    T21=imresize(T21,[39 60]);
    T22=imresize(T22,[39 60]);
    T23=imresize(T23,[39 60]);
    T24=imresize(T24,[39 60]);
    T25=imresize(T25,[39 60]);
    T26=imresize(T26,[39 60]);
    T27=imresize(T27,[39 60]);
    T28=imresize(T28,[39 60]);
    T29=imresize(T29,[39 60]);
    T30=imresize(T30,[39 60]);
    T31=imresize(T31,[39 60]);
    T32=imresize(T32,[39 60]);
    T33=imresize(T33,[39 60]);
    T34=imresize(T34,[39 60]);
```

```
        T35=imresize(T35,[39 60]);
        T36=imresize(T36,[39 60]);
        T37=imresize(T37,[39 60]);
        T38=imresize(T38,[39 60]);
        T39=imresize(T39,[39 60]);
        T40=imresize(T40,[39 60]);
        T41=imresize(T41,[39 60]);
        T42=imresize(T42,[39 60]);
        T43=imresize(T43,[39 60]);
        T44=imresize(T44,[39 60]);
        T45=imresize(T45,[39 60]);
        T46=imresize(T46,[39 60]);
        T47=imresize(T47,[39 60]);
        T48=imresize(T48,[39 60]);
        T49=imresize(T49,[39 60]);
        T50=imresize(T50,[39 60]);
        T51=imresize(T51,[39 60]);
        T52=imresize(T52,[39 60]);
        T53=imresize(T53,[39 60]);
        T54=imresize(T54,[39 60]);
        T55=imresize(T55,[39 60]);
        T56=imresize(T56,[39 60]);
        T57=imresize(T57,[39 60]);
        T58=imresize(T58,[39 60]);
        T59=imresize(T59,[39 60]);
        T60=imresize(T60,[39 60]);
    end
    T=zeros(39,60,60);
    T(:, :, 1) = T1;
    T(:, :, 2) = T2;
    T(:, :, 3) = T3;
    T(:, :, 4) = T4;
    T(:, :, 5) = T5;
    T(:, :, 6) = T6;
    T(:, :, 7) = T7;
    T(:, :, 8) = T8;
    T(:, :, 9) = T9;
    T(:, :, 10) = T10;
    T(:, :, 11) = T11;
    T(:, :, 12) = T12;
```

```
T(:, :, 13) = T13;
T(:, :, 14) = T14;
T(:, :, 15) = T15;
T(:, :, 16) = T16;
T(:, :, 17) = T17;
T(:, :, 18) = T18;
T(:, :, 19) = T19;
T(:, :, 20) = T20;
T(:, :, 21) = T21;
T(:, :, 22) = T22;
T(:, :, 23) = T23;
T(:, :, 24) = T24;
T(:, :, 25) = T25;
T(:, :, 26) = T26;
T(:, :, 27) = T27;
T(:, :, 28) = T28;
T(:, :, 29) = T29;
T(:, :, 30) = T30;
T(:, :, 31) = T31;
T(:, :, 32) = T32;
T(:, :, 33) = T33;
T(:, :, 34) = T34;
T(:, :, 35) = T35;
T(:, :, 36) = T36;
T(:, :, 37) = T37;
T(:, :, 38) = T38;
T(:, :, 39) = T39;
T(:, :, 40) = T40;
T(:, :, 41) = T41;
T(:, :, 42) = T42;
T(:, :, 43) = T43;
T(:, :, 44) = T44;
T(:, :, 45) = T45;
T(:, :, 46) = T46;
T(:, :, 47) = T47;
T(:, :, 48) = T48;
T(:, :, 49) = T49;
T(:, :, 50) = T50;
T(:, :, 51) = T51;
T(:, :, 52) = T52;
```

```
T(:, :, 53) = T53;
T(:, :, 54) = T54;
T(:, :, 55) = T55;
T(:, :, 56) = T56;
T(:, :, 57) = T57;
T(:, :, 58) = T58;
T(:, :, 59) = T59;
T(:, :, 60) = T60;

D1 =D;

N = 60;
for K = 1:N
    xx=imresize(T(:,:,K),[39 60]);
    %Target_data = D(K, :)';
    mynet=patternnet(100);
    % Set up Division of Data for Training, Validation,
Testing
    mynet.divideParam.trainRatio =100/100;
    mynet.divideParam.valRatio = 0/100;
    mynet.divideParam.testRatio = 0/100;
    mynet.trainParam.epochs=1000;
    mynet.trainParam.goal=1e-3;
    mynet.trainParam.lr=0.01;
    mynet.trainparam.min_grad=1e-9; %Diff. between
epochs
    % Train the Network
    net=train(mynet,xx,D1);
    %Calculating the error (MSE)
    error=sum(((vec2ind(net(xx)))-(vec2ind(D1))).^2)
    a=net(xx) %To show the prob. values of the network
output
end
plotconfusion(D1,a)
%[net error] = Neural2(T, D)
```