إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

نظام تعرف على كلمات عربية منفصلة يعتمد على المتحدث

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's name:	اسم الطالب: عامر محمد عامر الكرد
Signature:	التوقيع:
Date:	التاريخ: 2014/2/22

Islamic University, Gaza, Palestine
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department

Arabic Isolated Word Speaker Dependent Recognition System

By

Amer M. Elkourd

Supervisor

Prof. Ibrahim S. I. Abuhaiba

A Thesis Submitted in partial fulfillment of the Requirements for the degree of Master of Science in Computer Engineering

1435H (2014)

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي 1150

الرقم.. ج.س.غ/35/..... Ref

التاريخ 2014/02/22 Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **عامر محمد عامر الكرد** لنيل درجة الماجستير في **كلية *الهندسة*** قسم **هندسة الحاسوب** وموضوعها:

نظام تعرف على كلمات عربية منفصلة يعتمد على المتحدث

Arabic Isolated Word Speaker Dependent Recognition System

وبعد المناقشة التي تمت اليوم السبت 21 ربيع آخر 1435هـ، الموافق 2014/02/22م الساعة الحادية عشرة صباحًا، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

أ.د. إبراهيم سليمان أبو هيبة	مشرفاً ورئيساً	.....
د. وسام محمود عاشور	مناقشاً داخلياً	.....
د. إيهاب صلاح زقّوت	مناقشاً خارجيًا	.....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية *الهندسة*** / قسم **هندسة الحاسوب**.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب. 108 الرمال، غزة، فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps

Dedication

I wish to dedicate this thesis

To my parents, my brothers and my sisters

for their encouragements and supports.

Acknowledgment

First and foremost I would like to thank God for his generous blessings and for giving me strength and ability to complete this thesis.

Also, I wish to express my deepest appreciation and gratitude to my supervisor, Professor Ibrahim S. I. Abuhaiba for his assistance, guidance, advice and patience.

Table of Contents

List of Abbreviations

A/D	Analog-To-Digital Converter
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition system
DC	Direct Current
DCT	Discrete Cosine Transformation
DDMFCC	Delta-Delta Mel-Frequency Cepstral Coefficients
DFT	Discrete Fourier Transformation
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
EM	Expectation-Maximization
FFT	Fast Fourier Transformation
FIR	Finite Impulse Response Filter
GMM	Gaussian Mixture Model
GUI	Graphic User Interface
HMM	Hidden Markov Model
IIR	Infinite Impulse Response Filter
LIN	Linear Input Networks
LP	Linear Predictor
LPC	Linear Predictive Coding
LPCC	Linear Predictor Cepstral Coefficients
LVQ	Learning Vector Quantization
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficients
ML	Maximum Likelihood
MLP	Multilayer Perceptron
MRENN	Modular Recurrent Elman Neural Networks
PCM	Pulse-Code Modulation
PLP	Perceptual Linear Prediction
STFT	Short Time Fourier Transform
SVM	Support Vectors Machine
VQ	Vector Quantization

List of Figures

List of Tables

الملخص

نظام تعرف على كلمات عربية منفصلة يعتمد على المتحدث

عامر محمد الكرد

في هذه الرسالة قمنا بتصميم نظام جديد للتعرف على كلمات عربية منفصلة يعتمد على المتحدث باستخدام دمج عدة طرق لاستخلاص السمات وعدة طرق للتصنيفات. حيث أن النظام يقوم بدمج مخرجات الطرق باستخدام قاعدة التصويت للأغلبية. النظام تم تصميمه باستخدام واجهة رسومية في ماتلاب وعلى جهاز لابتوب معالجه Core i3 -2.26 جيجاهيرتز. قاعدة البيانات عبارة عن تسجيل ل 40 كلمة في وسط هادئ من قبل 5 أشخاص باستخدام ميكرفون اللابتوب، كل متحدث يقرأ الكلمة 8 مرات. خمسة منها ستستخدم في التدريب والباقي في التصنيف.

اولا في مرحلة المعالجة الاولية استخدمنا تقنية تحديد بداية ونهاية الكلمة وازالة الصمت باستخدام الطاقة ومعدل المرورالصفري (Zero Crossing Rate) . ولتنقية الكلمات من الضجيج استخدمنا تقنية تحويل المويجات المتقطع (Wavelet Transform Discrete). لزيادة سرعة النظام قمنا بجعل النظام يتعرف على المتحدث و يستخدم فقط قواعد البيانات الخاصة به.

قمنا بمقارنة 5 طرق مختلفة وهي: طريقة المسافة الايكلودية مع تقنية معاملات نغمة طيف التردد(Mel-Frequency Cepstral Coefficients) ، طريقة انحراف الوقت الديناميكي(Dynamic Time Warping) مع صفات الصوت (Formants) ، طريقة مطابقة القوالب (Gaussian Mixture) مع تقنية معاملات نغمة طيف التردد ، طريقة انحراف الوقت الديناميكي مع تقنية معاملات نغمة طيف التردد ، طريقة المسافة باستخدام تقنية Itakura مع الترميز التنبئي الخطي (Linear Predictive Coding). وقد وجدنا ان دقة النظام هي 85.23%، 57%، 87%، 90%، 83% بالترتيب. ولتحسينها قمنا باختبار حالات لدمج هذه الخمس طرق. ووجدنا ان افضلها كان بدمج MFCC|Euclidean + Formant|DTW + MFCC|DTW + LPC|Itakura حيث اصبحت الدقة 94.39%، غير أن زمن تنفيذها كان كبيرا حوالي 2.9 ثانية.

ولتقليل هذا الزمن قمنا باختبار دمج طريقتين كل مرة وفي حالة تعارضها نقوم باضافة باقي الطرق للنظام فوجدنا أفضلها من حيث الأداء وزمن التنفيذ كان بدمج MFCC | Euclidean + LPC | Itakura وعند عدم تطابقها يقوم النظام باضافة Formant | DTW + MFCC | DTW للتعرف على الكلمة. حيث أن متوسط زمن التنفيذ تقلص الى النصف حوالي 1.56 ثانية. ودقة النظام تحسنت لتصبح 94.56%.

أخيرا، النظام المقترح يعتبر جيد ومنافس مقارنة بابحاث سابقة.

Abstract

Arabic Isolated Word Speaker Dependent Recognition System

AMER M. ELKOURD

In this thesis we designed a new Arabic isolated word speaker dependent recognition system based on a combination of several features extraction and classifications techniques. Where, the system combines the methods outputs using a voting rule. The system is implemented with a graphic user interface under Matlab using G62 Core I3/2.26 Ghz processor laptop. The dataset used in this system include 40 Arabic words recorded in a calm environment with 5 different speakers using laptop microphone. Each speaker will read each word 8 times. 5 of them are used in training and the remaining are used in the test phase.

First in the preprocessing step we used an endpoint detection technique based on energy and zero crossing rates to identify the start and the end of each word and remove silences then we used a discrete wavelet transform to remove noise from signal. In order to accelerate the system and reduce the execution time we make the system first to recognize the speaker and load only the reference model of that user.

We compared 5 different methods which are pairwise Euclidean distance with Mel-Frequency cepstral coefficients (MFCC), Dynamic Time Warping (DTW) with Formants features, Gaussian Mixture Model (GMM) with MFCC, MFCC+DTW and Itakura distance with Linear Predictive Coding features (LPC) and we got a recognition rate of 85.23%, 57% , 87%, 90%, 83% respectively. In order to improve the accuracy of the system, we tested several combinations of these 5 methods. We find that the best combination is MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura with an accuracy of 94.39% but with large computation time of 2.9 seconds.

In order to reduce the computation time of this hybrid, we compare several subcombination of it and find that the best performance in trade off computation time is by first combining MFCC | Euclidean + LPC | Itakura and only when the two methods do not match the system will add Formant | DTW + MFCC | DTW methods to the combination, where the average computation time is reduced to the half to 1.56 seconds and the system accuracy is improved to 94.56%.

Finally, the proposed system is good and competitive compared with other previous researches.

Keywords: Arabic Speech recognition, Isolated Word, MFCC , FORMANTS , LPC, GMM , DTW, DWT, Euclidean, Itakura, Hybrid system.

Chapter 1
Introduction

1.1. Speech Recognition

Automatic Speech Recognition system (ASR) is used to convert spoken words into text. It has very important applications such as command recognition, dictation, foreign language translation, security control (verify the identity of the person to allow access to services such as banking by telephone). ASR makes writing on computers applications much easier and faster than using keyboards and could help handicapped people to interact with society. Also, it could be used to remotely turn on/off the home lights and electrical appliances.

ASR has two main types Discrete Word Recognition Systems and Continuous Speech Recognition Systems; and each type can be further subdivided into two categories as Speaker Dependent and Speaker Independent. Speaker dependent speech recognition systems operate only on the speech of a particular speaker for which the system is trained while the Speaker Independent Systems can be operated on the speech of any speaker.

Automatic Speech Recognition systems have two phases:

- A training phase during which the system learns the reference patterns representing the different speech sounds
- A recognition phase during which an unknown speech signal is identified using the stored reference patterns.

Figure 1.1 shows the block diagram of a speech recognition system. It consists of:

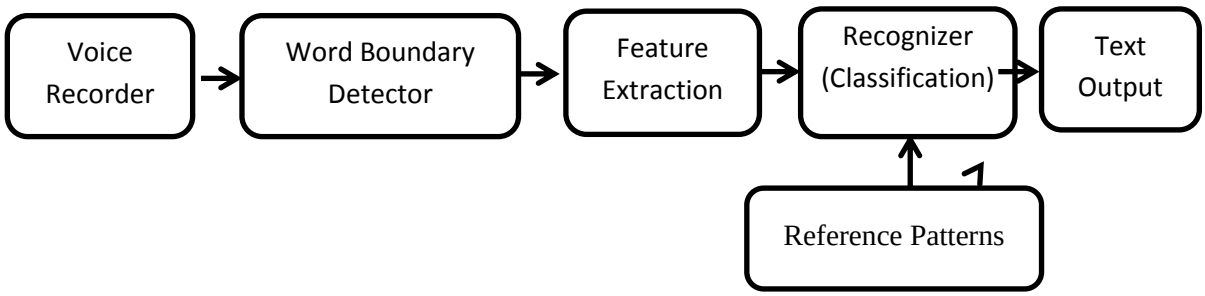

Figure 1.1 Speech recognition system block diagram

1.2. Word Boundary Detection

Word boundary detection is used to automatically identify the words and remove the silence at the beginning and the end of the input signal. Accurate word boundary detection improves the accuracy of the speech recognition system and reduces the amount of processing. The main techniques used in word boundary detction are:

- Short time energy.
- Short time zero crossing rate.
- Short time pitch frequency.
- The combination of energy and the zero crossing rate thresholds.

1.3. Front-End Analysis (Feature Extraction)

Feature extraction is the first step in an automatic speech recognition system. It aims to extract features from the speech waveform that are compact and efficient to represent the speech signal.

Since speech is a non-stationary signal. The feature parameters should be estimated over short-term intervals from 10ms to 30ms, in which Speech is considered to be stationary.

The major types of front-end processing techniques are:

- Linear Predictive Coding (LPC)
- Mel-Frequency cepstral coefficients (MFCC) +first and/or second derivatives of MFCC (delta and delta-delta coefficients).
- Perceptual Linear Prediction (PLP)
- Energy and zero crossing rate

MFCC is the most used feature extraction technique.

1.3.1. Mel-Frequency Cepstral Coefficients (MFCC)

MFCC is based on a perceptually scaled frequency axis. The mel-scale provides higher frequency resolution on the lower frequencies and lower frequency resolutions on higher frequencies. This scaling is based on hearing system of human ear. MFCC algorithm is shown in Figure 1.2.

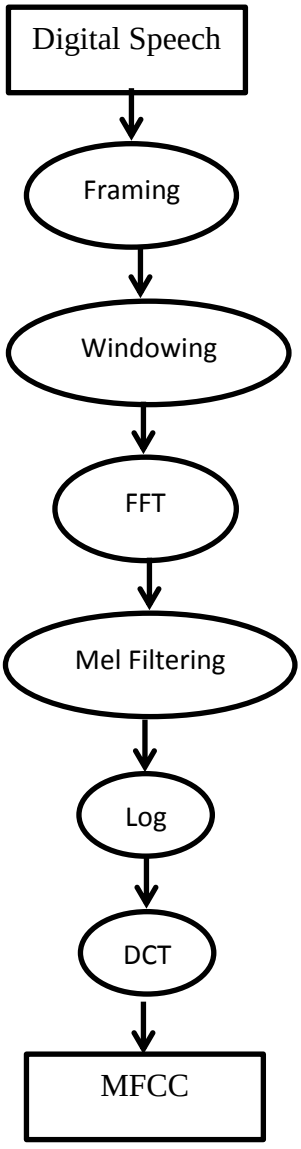

Figure 1.2 Block diagram of MFCC algorithm

The first step of the MFCC algorithm is called "framing". At this step the time interval for the feature extraction is determined. Generally a frame length of 10ms to 30ms is chosen for speech recognition. Overlapped framing is used for effective information extraction between two adjacent frames. That means, for example, a frame of 30ms is shifted 10ms to have a new frame, 20ms of previous frame is included in new one.

In "windowing" step a window function is applied to the frame. "Hamming" window is the most frequently used windowing technique for speech processing. It is defined by the following formula:

$$w(n) = 0.54 - 0.46\cos\frac{2\pi n}{N-1} \quad , \quad 0 \leq n \leq N-1 \qquad (1.1)$$

Where: N is the length in frame of the window and n is the frame index.

The Fast Fourier Transformation (FFT) is then applied to the window to have the frequency content of speech signal in current frame. The frequencies are then filtered by Mel-scale filter that imitates the varying resolution of the human ear with frequency filters spaced linearly at low frequencies and logarithmically at high frequencies and which is defined as:

$$Mel(f) = 2595\log_{10}(1 + \frac{f}{700}) \qquad (1.2)$$

Where: f is the frequency in Hz.

The Discrete Cosine Transformation (DCT) is applied to the logarithm of the mel-scale filtered frequencies. The first N coefficients (usually 13) are selected as feature vector representing the selected frame [1].

1.4. Acoustic Pattern Recognition Block

A pattern classification measure the similarity between an input speech and a reference pattern or a model obtained during training and accordingly determines a reference or a model, which best matches the input speech, as an output. It has four known approaches:

1. Template matching,
2. Statistical classification,
3. Syntactic or structural matching,
4. Neural networks

These methods are not necessarily independent from each other; it is possible to combine two or more of them to obtain a new and more performant classification method.

1.4.1. Template Matching

Template matching is the simplest and earliest approach to pattern classification.
Matching is used to determine the similarity between two observations of the same type.
In template matching, a template of pattern to be recognized is already available to the system. The pattern to be matched is compared with the stored template according to some distance (similarity) measure. This measure should be aware of scale changes, rotations or translations. Stored templates can be optimized with some training data before they are used for classification. When the number of classes increase and intra-class variability is high the performance of this method decreases.

Dynamic Time Warping method (DTW) is an example of template matching method. It creates templates (feature vectors) for each class and makes time alignment when comparing two realizations of same word since a two pronunciations of same word in different times are not the same because of speed and duration changes in the speech [1].

1.4.2. Statistical Approach

Statistical approach is based on features. Each pattern is represented as a feature vector. Given a set of training feature vectors, the purpose is to classify the feature vector into pre-defined class. Each class must be represented by sufficient amount of feature vectors in training data set. The class boundaries are determined statistically by probability distributions of the pattern belonging to each class. The performance of this method depends on good representation of each class in training data with a sufficiently large database which cover all intra-class variations that may be present in each class. The performance of class boundary determination algorithm is also important for better classification results. Hidden Markov Model (HMM) is an example of a statistical classification method [1].

1.4.3. Syntactic Approach

Syntactic approach is based on a hierarchical processing of subpatterns. Each pattern is composed of subpatterns and subpatterns are composed of simpler structures. There is a formal analogy between the syntax of the language which created the pattern and the structure of pattern. This method requires large amount of data to train the grammar for each pattern [1].

1.4.4. Neural Networks

Neural networks attempt to use some organizational principles (learning, generalization, computation, etc.) in a network of weighted directed graphs. They are capable of learning complex nonlinear relationships between output and input through weight updating of graph nodes (neurons). Feed-forward network and multi-layer perceptron are commonly used neural networks for pattern classification tasks. The first step of pattern classification with neural networks is, as with the other pattern classification methods, training which is called learning in this context. In this step network weights are updated to have minimum classification errors according to some pre-classified training data [1].

ASR is still a challenging task; its performance is still far below the human one and the accuracy of current recognition systems is not sufficient especially the Arabic ones.
Although Arabic is currently one of the most widely spoken language in the world, there has been relatively little speech recognition research on Arabic compared to the other languages [2, 3, 4].

Speech production is a complicated process. Even though people may sound alike to the human ear, everybody, to some degree, have a different and unique annunciation in their speech. Even the same speaker cannot produce the same utterance twice. Moreover, speech can be distorted by noise due to background noise, noise generated by microphones or different background environment during training and testing as well as emotional and the physical conditions of an individual.

Speech variation are due to speaking style, speaking rate, gender, age, accent, environment, health condition, prosody, emotional state, spontaneity, speaking effort, dialect ,articulation effort, …etc.

Feature extraction is a critical problem to get an accurate speech recognition system [5]. The recognition performance heavily depends on the performance of the feature extraction. Thus choice of features and its extraction from the speech signal should be such that it gives high recognition performance with reasonable amount of computation [6].

1.5. Research Overview

In this section, we present detailed information about this thesis. First, we start by identifying the motivations behind this study, objectives to be accomplished, methodology that has been followed, and our contributions throughout this work and finally, we show the content of this research.

1.5.1. Motivation

- Though Arabic language is a widely spoken language, research done in the area of Arabic Speech Recognition is limited when compared to other similar languages.
- Speech Recognition accuracies are still far away from a 100% recognition of natural human speech.
- Arabic speech recognitions are still poor and need improvement in recognition accuracy.
- The critical problem in developing highly accurate Arabic speech recognition systems is the choice of feature extraction and classification techniques. Currently, most of the speech recognition system use Mel Frequency Cepstral Coefficients (MFCCs) and Hidden Markov Models (HMM) in classification.

- System combination is one of the emerging techniques that can improve speech recognition accuracy and will take an important role in future speech technology research.
- A very rare research in Arabic recognition has tried combination of features and classification approach.
- Using a hybrid of classification methods can combine the advantages of these pattern recognition techniques and improve the speech recognition.

Hence, we are motivated to design a new Arabic speech recognition system based on new features combination and classification methods in order to improve the accuracy of the recognition.

1.5.2. Objectives

The goals of this thesis are:

- To design an Arabic speech recognition system with high accuracy rate.
- To find a new combination of features extractions and classification that improves the accuracy rate of an Arabic speech recognition system.
- The combination should be with different features and different information content (not for example MFCC feature and its derivatives) in order to find an optimal way of utilizing the mutually complementary classification information of different features.

1.5.3. Methodology

This research is carried out in different stages as described below:

- Record Arabic isolated speech words from different speakers to be used in training and testing.
- Test several features extractions methods on speech samples and find the most suitable algorithms.
- Find best features combinations that have the highest accuracy rate.

- Train the features vectors using the recorded datasets, in order to build the reference model.
- Test several classification methods using the test features and find the best classification hybrid that improves the accuracy.
- Evaluate the performance of the recognition system.

1.5.4. Thesis Contribution

This thesis aims to design an isolated word Arabic speech recognition system with high accuracy. The main contributions of this thesis include:

- We used in the preprocessing of the speech, a word boundary detector to automatically identify the words in the input signal by using the energy and the zero crossing rate.
- We applied discrete wavelet transform to the speech signal before extracting the features to improve the accuracy of the recognition and to make the system more robust to noise.
- We make a combination of several features extractions techniques : MFCC, Formants , Linear Predictive Coding features in order to improve the accuracy of the isolated word recognition system.
- We estimated the parameters of a Gaussian Mixture Model to fit the distribution of the training vectors. Then we classified the test sound to the Gaussian model that has the maximum posterior probability.
- We make hybrid of Gaussian Mixture Model, Template Matching with dynamic time wrapping and Pairwise Euclidean distances for the classification method.
- We make the code to first identify the speaker then load the training features of this person to be used in the recognition in order to increase the speed of recognition and its accuracy.

1.5.5. Outline of Rest of Thesis

This thesis is organized as follows:

Chapter 2; provides insight about related works in the field of Arabic speech recognition and the main used methods.

Chapter 3; provides overview of signal preprocessing, word boundary detection, features extractions techniques and the different classification methods.

Chapter 4; describes the proposed feature extraction and classification techniques in the system.

Chapter 5; describes the database in our experiments and the designed Matlab Graphic User Interface. Then, the results of these experiments will be shown and discussed.

Chapter 6; concluding remarks are stated. Future works, which may follow this study, are also presented.

Chapter 2
Related Work

2.1 Overview

Though Arabic is a widely spoken language, research done in the area of Arabic Speech Recognition is limited when compared to other similar languages.

This chapter presents an overview of the main features extraction and classification techniques in previous Arabic speech recognition researches, discusses their advantages, disadvantages and shows the benefits of the proposed method.

2.2 Feature Extraction Techniques

Feature extraction is the first step in an automatic speech recognition system. It aims to extract features from the speech waveform that are compact and efficient to represent the speech signal. The most famous features extraction techniques are:

1. Linear Predictive Coding (LPC).

2. Formants

3. Perceptual Linear Prediction (PLP).

4. Mel-Frequency Cepstral Coefficient (MFCC).

2.2.1. Linear Predictive Coding (LPC)

The Linear Predictive Coding presents a compact and precise representation of the spectral magnitude for signals and generates coefficients related to the vocal tract configuration. In LPC, speech sample can be estimated as a linear combination of past samples. Several researches have been performed on Arabic speech recognition using LPC features:

In [16], the authors designed a system using the scaly type architecture neural network for the recognition of speaker dependent isolated words for small vocabularies (11 words). They use LPC features extraction method and get a success rate of (79.5-88) %.

Choubassi et al. [10] implement Arabic isolated speech recognition. It uses Modular Recurrent Elman neural networks (MRENN) for recognition and LPC for feature extraction. The recognition rate for 6 Arabic words ranges between 85% and 100%.

Linear predictive coding (LPC) has always been a popular feature due to its accurate estimate of the speech parameters and efficient computational model of speech [24].

One main limitation of LPC features is the linear assumption that fails to take into account of the non-linear effects and sensitivity to acoustic environment and background noise [25].

Since our system will record dataset in calm environment then we will use LPC due to its advantages in non-noisy (silent) system.

2.2.2. Formants

The formants Fi represent the acoustic resonances produced by the dynamics of the vocal tract. It exhibits correlation with the production and perception of speech sounds. The formants depend on the shape of the mouth when producing sounds. The formant center-frequencies are usually given by the maximum amplitudes in the LPC spectrum of a speech sound which results from specific articulatory vocal tract settings.

Anissa et al. [13] designed a new Arabic isolated digit recognition system. They integrate some auxiliary features to improve the quality of recognition. They integrate: pitch frequency, energy and the first three formant frequencies and used HMM for classification. The system accuracy was between (59-97%).

Formants features are not used alone in speech recognition because formant frequencies cannot discriminate between speech sounds for which the main differences are unrelated to formants. Thus they are unable to distinguish between speech and silence or between vowels and weak fricatives.

But using formant in combination with other features improves the recognition. A speech recognizer employing formant features along with MFCCs is found to outperform the speech recognizer using only the conventional MFCCs in noisy conditions.

Also, formants are important in determining the phonetic content of speech and require small storage and can be computed quickly. Therefore, it will be selected in our combined feature system.

2.2.3. Perceptual Linear Predictive (PLP)

PLP technique is an auditory-like spectrum based on linear predictive analysis of speech. PLP combines several engineering approximations in modeling the psychophysical attributes of human hearing such as the critical band (Bark) frequency resolution, asymmetries of auditory filters, unequal sensitivity of human hearing at different frequencies, intensity-loudness non-linear relationship and broader than critical-band integration [25].

Park et al. [26] explored the training and adaptation of multilayer perceptron (MLP) features in Arabic ASRs. Three schemes had been investigated. First, the use of MLP features to incorporate short-vowel information into the graphemic system. Second, a rapid training approach for use with the perceptual linear predictive (PLP) + MLP system was described. Finally, the use of linear input networks (LIN) adaptation as an alternative to the usual HMM-based linear adaptation was demonstrated.

PLP is affected by factors such as the recording equipment, the communication channel or additive noise. It is an all-pole model, like LPC, so its advantages and disadvantages are similar to those of LPC but PLP is more complex and computationally intensive.

For this reason we will use LPC feature extraction in our system.

2.2.4. Mel-Frequency Cepstral Coefficients (MFCC)

MFCC is one of the most popular feature extraction techniques used in speech recognition. It is based on the frequency domain of Mel scale for human ear scale.

Speech signal had been expressed in the Mel frequency scale, in order to capture the important characteristics of speech.

Bassam et al. [8] develop and implement an Arabic speech recognition system using Hidden Markov Model Toolkit and uses MFCC for feature extraction. In [17], the authors designed an isolated word recognizer with phoneme based HMM models and MFCC features. The overall system performance of the 10 word digits was 93.72%. MFCC feature extraction is one of the more popular parameterization methods used by researchers in the speech technology field. It has the benefit that it is capable of capturing the phonetically important characteristics of speech. It gives a good discrimination and a small correlation between components.

A small drawback is that MFCC is not robust enough in noisy environments and combining it with other features such as formants improve its accuracy.

Finally, in our system we will try to find the best combination among MFCC, Formants and LPC that will give the best accuracy.

2.3 Classification Techniques used in Speech Recognition

The main important classification methods in speech recognition are:

i) Template approach.

ii) Statistical approach.

iii) Neural Network approach.

iv) Hybrid systems involving multiple models.

2.3.1 Template Based Approach:

Template based approach is one of the simplest and earliest approaches. It determines the similarity between unknown spoken word with each reference object in the training data and selecting the word with smallest distance. The major pattern recognition techniques for speech recognition are template method and Dynamic Time Warping method (DTW). Usually we use word as the smallest unit and templates for entire words are constructed. This has the advantage that, errors due to segmentation or classification of smaller acoustically more variable units such as phonemes can be avoided. In turn, each word must have its own full reference template which increasing memory size and computation as the number of vocabulary increases beyond a few hundred words. Template Matching performs speech recognition through a similarity measure, such as Euclidean distance, Dynamic Time Warping, k-Nearest Neighbor [23].

Hagos [18] designed a speaker-independent isolated-word Arabic digits recognizer that used Template Matching for input utterances. His system is based on the LPC parameters for feature extraction and log likelihood ratio for similarity measurements. Abdullah [19] developed another isolated-word Arabic digits recognizer that used positive-slope and zero-crossing duration as the feature extraction algorithm and template matching for classification. He reported 97% accuracy rate. Zaid et al. [27] designed a real-time Arabic speech recognition system using Matlab environment. They used MFCC as feature

extraction technique and in the recognition phase they used Euclidean distance and get a recognition rate of 89.6%.

An intrinsic advantage of template based recognition is that we do not need to model the speech process. This is very convenient, since our understanding of speech is still limited, especially with respect to its transient nature. Also, it has low error rates for distinctive words in speaker dependent isolated word recognition, and has simple programming requirements. Limitation of this technique, when similar words are included in the vocabulary, recognition performance can drop drastically. Also, it has recognition problems when used with large vocabularies, speaker-independence and continuous speech.

Since our system is a speaker dependent with a small dataset (40 distinct words) then the template approach will be a good choice. In the similarity measure, we will use two distance methods: Euclidean and Dynamic Time Warping (DTW). Where Euclidean distance is a simple and fast algorithm and it is one of the most commonly used distance measures. Also, Dynamic Time Warping is widely used in the small-scale speech recognition systems. It is used to measure the similarity between two words which may vary in time to cope with different speaking speeds. Additionally, the training procedure in DTW is very simple and fast, as compared with the HMM and ANN.

2.3.2 Statistical Approach

a. Hidden Markov Models (HMMs)

The Hidden Markov Models (HMMs) is widely used in speech recognition system. It covers from isolated speech recognition to very large vocabulary unconstrained continuous speech recognition. HMM is based on probabilistic models where the system is modeled as Markov process which can be represented as a state machine with unknown parameter through it. The main important process is to determine the unknown parameter. In [8], the authors designed an Arabic speech recognition system using MFCC features and HMM classification. Yousef et al. [17] designed an isolated digit recognizer using HMM models and MFCC features with accuracy of 93.72%. HMM is based on mathematical theorems which are useful for speech recognition. Also, it can easily be extended. New words can be added without affecting learnt HMMs. The major limitation

is that they work well only when the assumptions are satisfied and need to set huge number of parameters and require large amount of training data.

b. Gaussian Mixture Model (GMM)

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. The probability density functions of many random processes, such as speech, are non-Gaussian. A non-Gaussian pdf may be approximated by a weighted sum (i.e. a mixture) of a number of Gaussian densities of appropriate mean vectors and covariance matrices. GMM is a special case of an HMM, by assuming independent and identically-distributed consecutive frames.

N.Hammami et al. [28] designed an automatic recognition of the Spoken Arabic Digits based on (GMM) classifier and used Delta-Delta Mel-frequency cepstral coefficients (DDMFCC) for features extraction. They achieved 99.31% correct digit recognition.

Gaussian mixture model (GMM) is a conventional method for speech recognition, known for its effectiveness and scalability in speech modeling. GMM is simple, easy to evaluate and learn and faster to compute. Also, it is a class based training. In other words, adding new class to the database is done without retraining the whole system. The limitation of GMM that it requires a sufficient amount of training data to ensure good performance which increase the training time.

GMM is very competitive when compared to other pattern recognition techniques. It is more simple and faster than HMM with very small or no performance degradation.

For that we will use GMM in our system instead of HMM because in our hybrid system we need to combine fast and accurate approaches in order to have an acceptable computational time of the combinations.

2.3.3 Artificial Neural Network (ANN)

Neural networks are basically a dense interconnection of simple, nonlinear, computational elements. It attempts to use some organizational principles (learning, generalization, computation, etc.) in a network of weighted directed graphs. They are capable of learning complex nonlinear relationships between output and input through

weight updating of graph nodes (neurons). Feed-forward network and multi-layer perceptron are commonly used neural networks for pattern classification tasks.

Akram et al. [16] designed a speaker dependent isolated words system using LPC features and neural network classification. The best accuracy of the system was 88 %. Choubassi et al. [10] designed an Arabic isolated word speech recognition using LPC features and Modular Recurrent Elman neural networks in classification. The system accuracy was between 85% and 100%.

Neural networks are data driven and self-adaptive-learning and when an element of the neural network fails, it can continue without any problem by their parallel nature. Neural networks limitations are in requiring large dataset and resources to obtain good accuracy. Also, it has complex network structure and the individual relations between the input variables and the output variables are not developed by engineering judgment so that the model tends to be a black box and estimating its parameters to fit certain dataset may not be as good for recognition of other datasets. For this reasons neural networks approach will not be used in our system especially because we used small dataset.

2.3.4 Hybrid Methods

Hybrid methods try to reduce their limitations by combining the advantages of the combined techniques. It is one of the emerging approaches that can improve speech recognition accuracy and will take an important role in future speech technology research. Very few Arabic speech recognition researches have used a hybrid system. We can mention some of relevant researches:

a. Feature Combinations

Moustafa et al. [9] designed speaker-independent natural Arabic speech recognition system based on Mellon university Sphinx HTK tools. It uses HMM and a combination of features MFCC, differenced MFCC, 2nd order differenced MFCC, normalized power, differenced power and 2nd order differenced power. They used 14232 words in the training datasets. The limitation of their system is in combining similar features. Also, their system requires very large dataset to get a good result and they used only one classification method.

Shoaib et al. [12] presented a novel approach to develop a robust speaker independent Arabic phoneme identification system based on a hybrid set of speech features. This hybrid set consists of intensity contours and formant frequencies and used generalized regression neural network for classification .It was observed through extensive validation runs that the highest level of accuracy achieved was 82.59%. They found that combining features improves the accuracy. Their system has good result in letter recognition but not sure for word case. Also they used large number of speakers. They recorded sounds of 40 different speakers. Also, they did not use in the combination a cepstral features and they used only one classification method.

Abdullah [19] designed an Arabic digits recognizer using positive-slope and zero-crossing duration feature extraction and template matching for classification. Their system could be good for digit recognition but not sure for more words especially that he used weak features.

b. Combined Classifier

In [15, 21], the authors designed a system for recognition of isolated Arabic words by using a combined classifier. A combined classifier is based on a number of Back-Propagation/LVQ neural networks with different parameters and architectures and MFCC features are used. The datasets are records taken from the Holy Quran for many famous reciters. They recorded 10 words for each of 10 speakers with 6 repetitions for each word. The whole set of 600 words is divided into 300 utterances for training and 300 utterances for testing. For the unseen test set, the recognition rate of the Back-Propagation combined classifier was 96% and that of the LVQ combined classifier was 86.6%. The best individual classifiers resulted in 93% correct classification. They found that the implemented combined classifier outperforms those traditional classifiers which use the HMM-based speech recognition approaches.

The proposed system used only single feature extraction method MFCC and is tested only for 10 words and they restricted the type of words to not have Homophone or noise and they used famous Quran reciters not normal speaker. Also, they used manual segmentation to find the boundary of words and when classifying the training data they found some errors. Also, they used same type of classification method "neural network"

only and changing parameters between them. Also, using neural network in classification is time consuming and need large number of training data to get accurate results.

Anissa et al. [13] designed an Arabic isolated digit recognition using HMM classification and features combination of pitch, energy, and formant. The accuracy rate was between (59-97%). Their system is tested only for digit and some auxiliary features when added to a recognition system could degrade its performance and using combined classifier make these auxiliary features effective and improves the accuracy [29].

c. Hybrid System with Features Combinations and Combined Classifier

Very few researches have used features combinations with a combined classifier.

Bourouba et al. [11] presented a new arabic digit recognition system based on classifier combination of HMM and a supervised classifier (SVM or KNN) with MFCC and the log energy and pitch frequency feature extraction combination method .They found that using HMM classifier alone the accuracy is 88.26% and improved with the combined system to 92.72%. The limitation of their system is in using weak features and combined two slow classification methods.

The proposed technique in our system is to test a new combined classifier with features combination. We will find the best combination of formant and LPC and MFCC features along with a testing the combination of Euclidean and DTW and GMM for classification approach. Also, we need to find the best accuracy of the hybrid system in trade off the computation times.

Chapter 3
Background

A speech signal can be divided into two broad categories: periodic signals and the non-periodic signals with the form of a noise. Periodic signals are produced when there are vibrations of the vocal cords, while the non-periodic signals are produced when the air passes freely through the vocal tract. The periodicity of a voiced sound is determined by the frequency of vibration of the vocal cords. This frequency is called the fundamental frequency, it can vary:

- From 50 to 200 Hz for a male voice,

- From 150 to 450 Hz for a female voice,

- From 200 to 600 Hz for a child's voice.

The human vocal apparatus is like a filter. It amplifies frequencies near its resonant frequencies and attenuates the others. For each voiced sound, certain frequencies are amplified and others are attenuated. Amplified frequencies are named formants. These formants are in fact frequencies of resonance of the human vocal apparatus.

3.1. Pre-processing

Preprocessing is the fundamental signal processing applied before extracting features from speech signal, for the purpose of enhancing the performance of feature extraction algorithms. Commonly used preprocessing techniques include DC component removal, preemphasis filtering, and amplitude normalization.

3.1.1. DC Component Removal

The initial speech signal often has a constant component, i.e. a non-zero mean. This is typically due to DC bias within the recording instruments. The DC component can be easily removed by subtracting the mean value from all samples within an utterance.

DC offset occurs when hardware, such as a sound card, adds DC current to a recorded audio signal. This current produces a recorded waveform that is not centered on the baseline. Therefore, removing this DC offset is the process of forcing the input signal mean to the baseline [39]. An illustrative example of removing DC offset from a waveform file is shown in Figure 3.1.

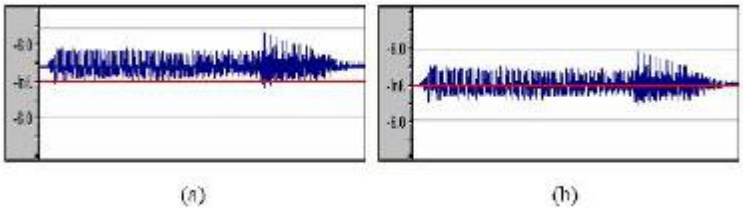

Figure 3.1 Removal of DC offset from a Waveform. (a) Exhibits DC offset, (b) After the removal of DC offset [39].

3.1.2. Preemphasis Filtering

A pre-emphasis filter compresses the dynamic range of the speech signal's power spectrum by flattening the spectral tilt. Typically, the filter is in form of

$$P(z) = 1 - az^{-1} \tag{3.1}$$

Where a: ranges between 0.9 and 1.0. A typical value of "a" is 0.95, which gives rise to a more than 20dB amplification of the high frequency spectrum [47].

The spectral slope of a human speech spectrum is usually negative since the energy is concentrated in low frequency. Thus, a preemphasis filter is introduced before applying feature algorithms to increase the relative energy of the high-frequency spectrum.

3.1.3. Amplitude Normalization

Recorded signals often have varying energy levels due to speaker volume and microphone distance. Amplitude Normalization can cancel the inconsistent energy level between signals, thus can enhance the performance in energy-related features.

There are several methods to normalize a signal's amplitude. One of them is achieved by a point-by-point division of the signal by its maximum absolute value, so that the dynamic range of the signal is constrained between -1.0 and +1.0. Another commonly used normalization method is to divide each sample point by the variance of an utterance.

3.1.4. The Discrete Wavelet Transform

The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. The Wavelet Transform provides a time-frequency representation of the signal. It was developed to overcome the short coming of the Short Time Fourier Transform (STFT), which can also be used to analyze non-stationary signals. While STFT gives a constant resolution at all frequencies, the Wavelet Transform uses multi-resolution technique by which different frequencies are analyzed with different resolutions.

The DWT can be used for Multi Resolution Analysis. The given signal is decomposed into the approximation and detail coefficients. A given function f(t) satisfying certain conditions, can be expressed through the following representation:

$$f(t) = \sum_{j=1}^{L} \sum_{k=-\infty}^{\infty} d(j,k)\varphi(2^{-j}\,t - K) + \sum_{k=-\infty}^{\infty} a(L,K)\theta(2^{-L}\,t - K) \qquad (3.2)$$

Where $\varphi(t)$, is the mother wavelet and θ(t) is the scaling function. a(L,k) is called the approximation coefficient at scale L and d(j,K) is called the detail coefficient at scale j. The approximation and detail coefficients can be expressed as

$$a(L,K) = \frac{1}{\sqrt{2^L}} \int_{-\infty}^{\infty} f(t)\theta(2^{-L}\,t - K)dt \qquad (3.3)$$

$$d(j,K) = \frac{1}{\sqrt{2^j}} \int_{-\infty}^{\infty} f(t)\varphi(2^{-j}\,t - K)dt \qquad (3.4)$$

The DWT is computed by successive lowpass and highpass filtering of the discrete time-domain signal as shown in Figure 3.2.

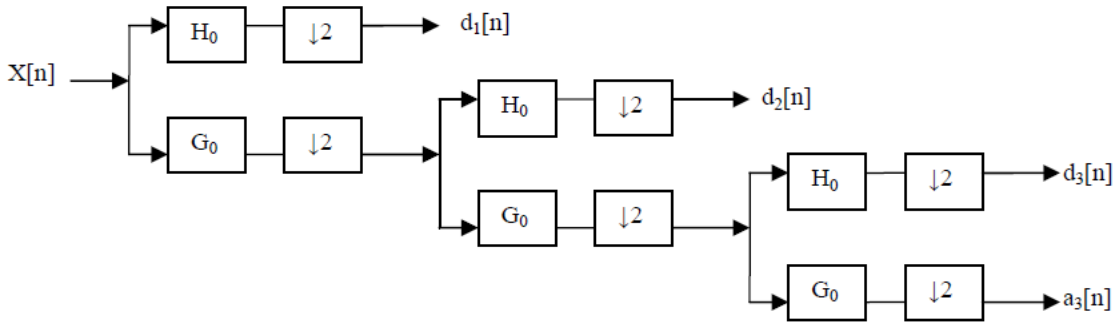

Figure 3.2 Three-level wavelet decomposition tree [38]

In the Figure 3.2, the signal is denoted by the sequence x[n], where n is an integer. The low pass filter is denoted by G0 while the high pass filter is denoted by H0. At each level, the high pass filter produces detail information; d[n], while the low pass filter associated with scaling function produces coarse approximations, a[n].

The filtering and decimation process is continued until the desired level is reached. The maximum number of levels depends on the length of the signal.

The DWT can be viewed as the process of filtering the signal using a low pass (scaling) filter and high pass (wavelet) filter. Thus, the first level of the DWT decomposition of a signal splits it into two bands giving a low pass version and a high pass version of the signal. The low pass signal gives the approximate representation of the signal while the high pass filtered signal gives the details or high frequency variations. The second level of decomposition is performed on the low pass signal obtained from the first level of decomposition. The wavelet decomposition of the signal S analyzed at level j has the following structure: [cAj, cDj, …, cD2, cD1].

There are a number of basis functions that can be used as the mother wavelet for Wavelet Transformation. Since the mother wavelet produces all wavelet functions used in the transformation through translation and scaling, it determines the characteristics of the resulting Wavelet Transform. Therefore, the details of the particular application should

be taken into account and the appropriate mother wavelet should be chosen in order to use the Wavelet Transform effectively [38].

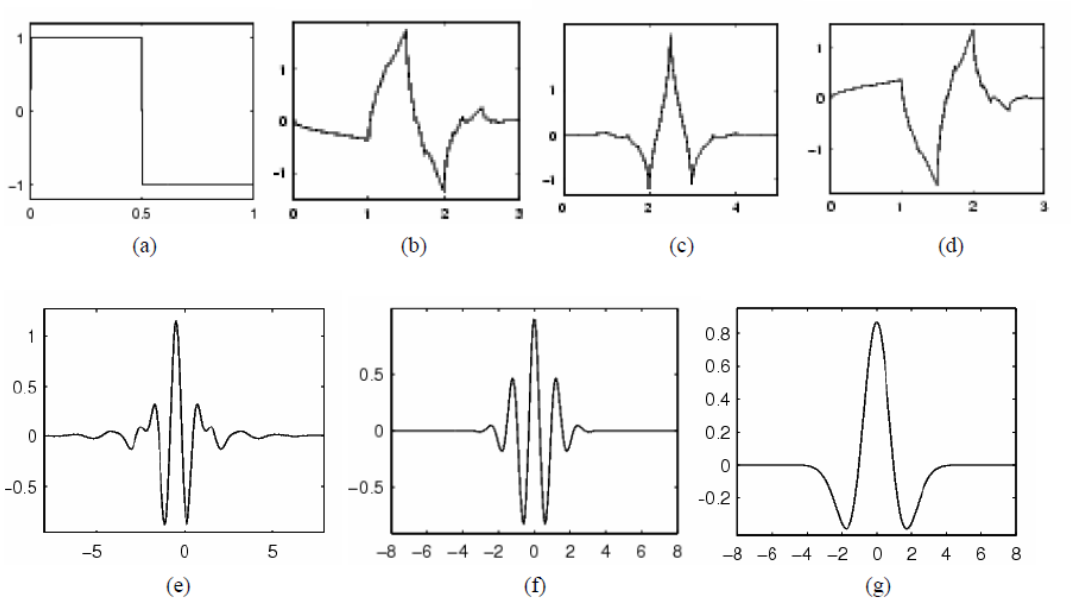

Figure 3.3 Wavelet families (a) Haar (b) Daubechies4 (c) Coiflet1 (d) Symlet2 (e) Meyer (f) Morlet (g) Mexican Hat [38]

Figure 3.3 illustrates some of the commonly used wavelet functions. Haar wavelet is one of the oldest and simplest wavelet. Therefore, any discussion of wavelets starts with the Haar wavelet. Daubechies wavelets are the most popular wavelets. They represent the foundations of wavelet signal processing and are used in numerous applications. These are also called Maxflat wavelets as their frequency responses have maximum flatness at frequencies 0 and π. This is a very desirable property in some applications. The Haar, Daubechies, Symlets and Coiflets are compactly supported orthogonal wavelets. These wavelets along with Meyer wavelets are capable of perfect reconstruction. The Meyer, Morlet and Mexican Hat wavelets are symmetric in shape. The wavelets are chosen based on their shape and their ability to analyze the signal in a particular application [38].

3.2. The Speech Endpoint Detection

The performance of speech recognition system is often degraded in adverse environments. Accurate Speech endpoint detection is very important for robust speech recognition.

The speech endpoint detection is aim to distinguish the speech and non-speech phase, it plays the key role in the speech signal processing. Inaccurate endpoint detection causes the recognition ratio lower and increases the computation amounts. The research shows that, the probability of false recognition of isolating word is over half cause by unreliable endpoint detection.

During the last decades, a number of endpoint detection methods have been developed. We can categorize approximately those methods into two classes. One is based on thresholds. Generally, this kind of methods first extracts the acoustic features for each frame of signals and then compares these values of features with preset thresholds to classify each frame. The other is pattern-matching method that needs estimate the model parameters of speech and noise signal. The method based on pattern-matching has the traits of high accuracy, but the disadvantages are model dependency, high complexity and enormous computation. It is difficult to apply for the real-world speech signal processing system. Compared with pattern-matching method, threshold-based method is simpler and faster since it does not need to keep much training data and train models. Traditional short-time energy and zero-crossing rate method is part of this sort, but it is sensitive to varies types of noise and cannot fully specify the characteristics of a speech signal, the detection effect will become worse in the adverse environment [40].

3.3. Feature Extraction

The goal of feature extraction is to represent any speech signal by a finite number of measures (or features) of the signal. This is because the entirety of the information in the acoustic signal is too much to process, and not all of the information is relevant for specific tasks. In present ASR systems, the approach of feature extraction has generally been to find a representation that is relatively stable for different examples of the same speech sound, despite differences in the speaker or environmental characteristics, while keeping the part that represents the message in the speech signal relatively intact. The

main features extractions techniques are: Linear Predictive Coding (LPC), Perceptual Linear Prediction (PLP) and Mel-Frequency Cepstral Coefficient (MFCC)

3.3.1. Mel-Frequency Cepstral Coefficient (MFCC)

MFCC is one of the most popular features extractions techniques used in speech recognition, whereby it is based on the frequency domain of Mel scale for human ear scale. MFCC is based on the known variation of the human ear's critical bandwidths with frequency. Speech signal had been expressed in the Mel frequency scale, in order to capture the important characteristics of phonetic in speech. This scale has a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. Normal speech waveform may vary from time to time depending on the physical condition of speakers'vocal cord. Rather than the speech waveforms themselves, MFCCs are less susceptible to the said variations. MFCC block diagram consist of the following steps:

1. Preprocessing.
2. Framing.
3. Windowing.
4. Discrete Fourier Transformation (DFT).
5. Mel-Filterbank.
6. Logarithm.
7. Discrete Cosine Transformation (DCT).

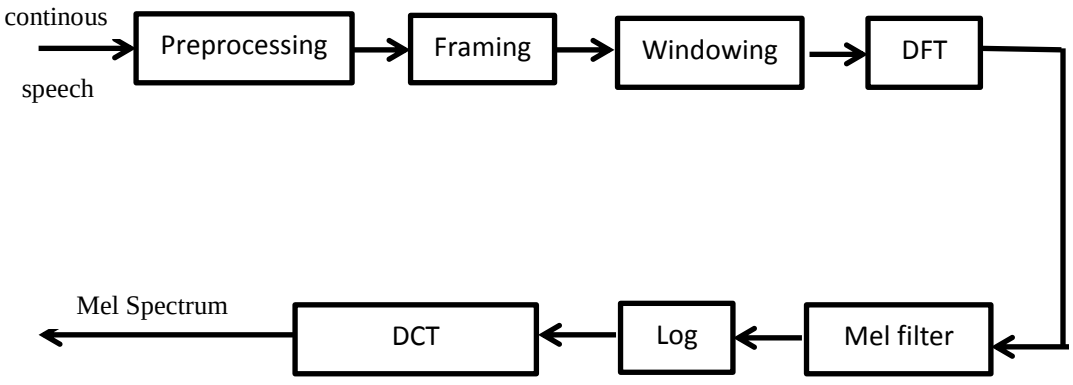

Figure 3.4 Block diagram of the computation steps of MFCC

3.3.2. Linear Predictive Coding (LPC)

LPC has been considered one of the most powerful techniques for speech analysis. LPC relies on the lossless tube model of the vocal tract. The lossless tube model approximates the instantaneous physiological shape of the vocal tract as a concatenation of small cylindrical tubes. The model can be represented with an all pole (IIR) filter. LPC coefficients can be estimated using autocorrelation or covariance methods.

The drawback of LPC may estimate the high sensitivity to quantization noise. By converting the LPC coefficients back into cepstral coefficient, it can decrease the sensitivity of high and low order cepstral coefficient to noise.

Linear prediction analysis characterizes the shape of the spectrum of a short segment of speech with a small number of parameters for efficient coding.

The basic idea behind linear predictive analysis is that a specific speech sample at the current time can be approximated as a linear combination of past speech samples. Through minimizing the sum of squared differences (over a finite interval) between the actual speech samples and linear predicted values a unique set of parameters or predictor coefficients can be determined. These coefficients form the basis for linear predictive analysis of speech.

We can predict that the nth sample in a sequence of speech samples is represented by the weighted sum of the p previous samples [41]:

$$\check{x}[n] = \sum_{k=1}^{p} a_k \, x[n-k] \qquad (3.5)$$

The number of samples (p) is referred to as the "order" of the LPC. As p approaches infinity, we should be able to predict the nth sample exactly. However, p is usually on the order of ten to twenty, where it can provide an accurate enough representation with a limited cost of computation.

Linear Predictive Coding (LPC) is one of the methods of compression that models the process of speech production. Under normal circumstances, speech is sampled at 8000 samples/second with 8 bits used to represent each sample. This provides a rate of 64000 bits/second. Linear predictive coding reduces this to 2400 bits/second. At this reduced rate the speech has a distinctive synthetic sound and there is a noticeable loss of quality.

However, the speech is still audible and it can still be easily understood. Since there is information loss in linear predictive coding, it is a lossy form of compression.

LPC determines the coefficients of a FIR filter that predicts the next value in a sequence from current and the previous inputs. This type of filter is also known as a one-step forward linear predictor. LP analysis is based on the all-pole filter described in Equation 3.6 [41]:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}} \tag{3.6}$$

Where: $\{a_k/(1\leq k\leq p)\}$ are the predictor coefficients and p is the order of the filter.

Transforming Equation 3.6 to the time-domain, as shown in Equation 3.7, predicts a speech sample based on a sum of weighted past samples.

$$s'(n) = \sum_{k=1}^{p} a_k \ .s(n-k) \tag{3.7}$$

Where s′ (n) is the predicted value based on the previous values of the speech signal s(n). LP analysis requires estimating the LP parameters for a segment of speech. The idea is to find a_k's, so that Equation 3.7 provides the closest approximation to the speech samples. This means that s′(n) is closest to s(n) for all values of n in the segment. The spectral shape of s(n) is assumed to be stationary across the frame, or a short segment of speech.

The error, e, between the predicted value and the actual value is

$$e(n)=s(n)-s'(n) \tag{3.8}$$

The summed squared error, E, over a finite window of length N is

$$E = \sum_{n=0}^{N+p-1} e^2(n) \tag{3.9}$$

The minimum value of E occurs when the derivative is zero with respect to each of the parameters a_k. By setting the partial derivatives of E, a set of p equations are obtained. The matrix form of these equations is

$$\begin{bmatrix} R(0) & R(1) & R(p-1) \\ R(1) & R(0) & R(p-2) \\ \cdots & \cdots & \cdots \\ R(p-1) & R(p-2) & R(0) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \cdots \\ R(p) \end{bmatrix} \quad (3.10)$$

Where R(i) is the autocorrelation of lag i computed as

$$R(i) = \sum_{m=0}^{N-1-i} s(m) \,.\, s(m+i) \quad (3.11)$$

N is the length of the speech segment s(n).

The Levinson-Durbin algorithm solves the nth order system of linear equations

$$R \;.\; a \;=\; b \quad (3.12)$$

For the particular case where R is a Hermitian, positive-definite, Toeplitz matrix and b is identical to the first column of R shifted by one element.

The autocorrelation coefficients R(k) are used to compute the LP filter coefficients ai, i=1…p, by solving the set of equations:

$$\sum_{i=1}^{r} a_i \;.\; r(|i-k|) = r(k) \quad (3.13)$$

Where k=1…p.

This set of equations is solved using the Levinson-Durbin recursion, Equation 3.14 through Equation 3.18.

$$E(0) = r(0) \quad (3.14)$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} a_j^{i-1} \;.\; r(i-j)}{E(i-1)} \quad (3.15)$$

$$a_i^{(i)} = k_i \quad (3.16)$$

$$j^{(i)} = a_j^{i-1} - k_i \;.\; a_{i-j}^{i-1} \quad (3.17)$$

$$E(i) = (1 - k_i^2) \,.\, E(i-1) \quad (3.18)$$

Where: $1 \le j \le i-1 \quad and \quad 1 \le i \le p$.

The parameters k_i are known as the reflection parameters. If the condition $|k_i| \leq 1$, where $1 \leq i \leq p$ is satisfied, the roots of the polynomial predictor all lie within the unit circle in the z-plane, and the all-pole filter is stable [41].

3.3.3. Formants

Formants characterize the "filter" portion of a speech signal. They are the poles in the digital resonance filter or digital resonator. Given the source-filter model for voiced speech that is free of nasal coupling, the all-pole filter is characterized by the pole positions, or equivalently by the formant frequencies, F1, F2, …, Fn, formant bandwidths, B1, B2, …, Bn, and formant amplitudes, A1, A2, …, An. Among them, the formant frequencies or resonance frequencies, at which the spectral peaks are located, are the most important. A formant frequency is determined by the angle of the corresponding pole in the discrete-time filter transfer function.

The normal range of the formant frequencies for adult males is F1 =180 - 800 Hz,

F2 =600 - 2500 Hz, F3 =1200 – 3500 Hz, and F4 =2300 - 4000 Hz. These ranges have been exploited to provide constraints for automatic formant extraction and tracking.

The average difference between adjacent formants adult males is about 1000 Hz. For adult females, the formant frequencies are about 20% higher than adult males. The relationship between male and female formant frequencies, however, is not uniform and the relationship deviates from a simple scale factor. When the velum is lowered to create nasal phonemes, the combined nasal+vocal tract is effectively lengthened from its typical 17 cm vocal-tract length by about 25%. As a result, the average spacing between formants reduces to about 800 Hz [37].

Formant bandwidths are physically related to energy loss in the vocal tract, and are determined by the distance between the pole location and the origin of the z-plane in the filter transfer function. Empirical measurement data from speech suggest that the formant bandwidths and frequencies are systematically related. Formant amplitudes, on the other hand, vary with the overall pattern of formant frequencies as a whole. They are also related to the spectral properties of the voice source [37].

LPC analysis models voiced speech as the output of an all-pole filter in response to a simple sequence of excitation pulses. In addition to major speech coding and recognition

applications, LPC is often used as a standard formant extraction and tracking method. It has limitations in that the vocal-tract filter transfer function, in addition to having formant poles (which are of primary interest for speech analysis), generally also contains zeros due to sources located above the glottis and to nasal and subglottal coupling. Furthermore, the model for the voice source as a simple sequence of excitation impulses is inaccurate, with the source actually often containing local spectral peaks and valleys. These factors often hinder accuracy in automatic formant extraction and tracking methods based on LPC analysis. The situation is especially serious for speech with high pitch frequencies, where the automatic formant-estimation method tends to pick harmonic frequencies rather than formant frequencies. Jumps from a correctly-estimated formant in one time frame to a higher or a lower value in the next frame constitute one common type of tracking error.

The automatic tracking of formants is not trivial. The factors rendering formant identification complex include the following. The ranges for formant center-frequencies are large, with significant overlaps both within and across speakers. In phoneme sequences consisting only of oral vowels and sonorants, formants smoothly rise and fall, and are easily estimated via spectral peak-picking. However, nasal sounds cause acoustic coupling of the oral and nasal tracts, which lead to abrupt formant movements. Zeros (due to the glottal source excitation or to the vocal tract response for lateral or nasalized sounds) also may obscure formants in spectral displays. When two formants approach each other, they sometimes appear as one spectral peak (e.g., F1-F2 in back vowels). During obstruent sounds, a varying range of low frequencies is only weakly excited, leading to a reduced number of formants appearing in the output speech.

Given a spectral representation S(z) via the LPC coefficients, one could directly locate formants by solving directly for the roots of the denominator polynomial in S(z). Each complex-conjugate pair of roots would correspond to a formant if the roots correspond to a suitable bandwidth (e.g., 100-200~Hz) at a frequency location where a formant would normally be expected. This process is usually very precise, but quite expensive since the polynomial usually requires an order in excess of 10 to represent 4-5 formants. Alternatively, one can use phase to label a spectral peak as a formant. When evaluating

S(z) on the unit circle a negative phase shift of approximately 180 degrees should occur as the radiant frequency passes a pole close to the unit circle (i.e., a formant pole).

Two close formants often appear as a single broad spectral peak, a situation that causes many formant estimators difficulty, in determining whether the peak corresponds to one or two resonances. A method called the chirp z-transform has been used to resolve this issue.

Formant estimation is increasingly difficult for voices with high F0, as in children's voices. In such cases, F0 often exceeds formant bandwidths, and harmonics are so widely separated that only one or two make up each formant. A spectral analyzer, traditionally working independently on one speech frame at a time, would often equate the strongest harmonics as formants. Human perception, integrating speech over many frames, is capable of properly separating F0 and the spectral envelope (formants), but simpler computer analysis techniques often fail. It is wrong to label a multiple of F0 as a formant center frequency, except for the few cases where the formant aligns exactly with a multiple of F0 (such alignment is common in song, but much less so in speech) [37].

3.4. Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

A Gaussian mixture model is a weighted sum of M component Gaussian densities as given by the equation:

$$p(x|y) = \sum_{i=1}^{M} \omega_i \; g(x|\mu_i, \Sigma_i) \tag{3.19}$$

Where: x is a D-dimensional continuous-valued feature vector.

wi, i = 1, . . . ,M, are the mixture weights, and g(x|μi,Σi), i = 1, . . . ,M, are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{2\pi^{D/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)' \ \Sigma_i^{-1}(x-\mu_i)\} \quad (3.20)$$

with, mean vector μi and covariance matrix Σ_i.

The mixture weights satisfy the constraint that:

$$\sum_{i=1}^{M} \omega_i \ = 1 \quad (3.21)$$

The covariance matrices in GMM can be full rank or constrained to be diagonal, but because the component Gaussian are acting together to model the overall feature density, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modeling the correlations between feature vector elements.

3.4.1. Maximum Likelihood Parameter Estimation

There are several techniques available for estimating the parameters of a GMM. By far the most popular and well-established method is maximum likelihood (ML) estimation.

The aim of ML estimation is to find the model parameters which maximize the likelihood of the GMM given the training data. For a sequence of T training vectorsX ={x1, . . . , xT}, the GMM likelihood, assuming independence between the vectors, it can be written as:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \quad (3.22)$$

ML parameter estimation can be obtained iteratively using the expectation-maximization (EM) algorithm.

The basic idea of the EM algorithm is, beginning with an initial model λ, to estimate a new model $\tilde{\lambda}$. Such that $p(X|\tilde{\lambda}) \geq P(X|\lambda)$. The new model then becomes the initial model for the next iteration and the process is repeated until some convergence threshold is reached.

Chapter 4
The Proposed System Solution

4.1 Data Collection

In the data collection stage we recorded 40 Arabic words with 5 different speakers (3 male and 2 female) using computer microphone with sampling frequency of 8 kHz, 16-bit PCM WAV format. Each speaker read every word 8 times (5 of them are used in training and the remaining are used in the test phase). The list of the words is shown in Table 4.1:

Table 4-1 List of words used in the system

1	أمام	21	ابدأ
2	خلف	22	توقف
3	يمين	23	اكمل
4	يسار	24	امسح
5	أعلى	25	احمل
6	أسفل	26	انظر
7	تحرك	27	انطلق
8	قف	28	اعد
9	أسرع	29	نعم
10	تمهل	30	لا
11	افتح	31	صفر
12	أغلق	32	واحد
13	انزل	33	اثنين
14	اصعد	34	ثلاثة
15	اقرأ	35	أربعة
16	اكتب	36	خمسة
17	تكلم	37	ستة
18	اسكت	38	سبعة
19	أجب	39	ثمانية
20	فوق	40	تسعة

4.2 Software

Two software programs are used during the development of the recognition system

- MATLAB R2010a: is used in writing the code of the system. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
- Praat software: is used in voice editing and spectrum analysis of the collected data.

4.3 System Block Diagram

The speech recognition system consists of two stages, a training stage and a recognition stage both stages have common blocks which are wave recording, speech pre-processing, word boundary detection and features extraction. The output of the training stage is a reference model.

In the recognition stage the extracted features are compared with the reference model and the word that has the best match will be the output. Figure 4.1 shows the block diagram of the System.

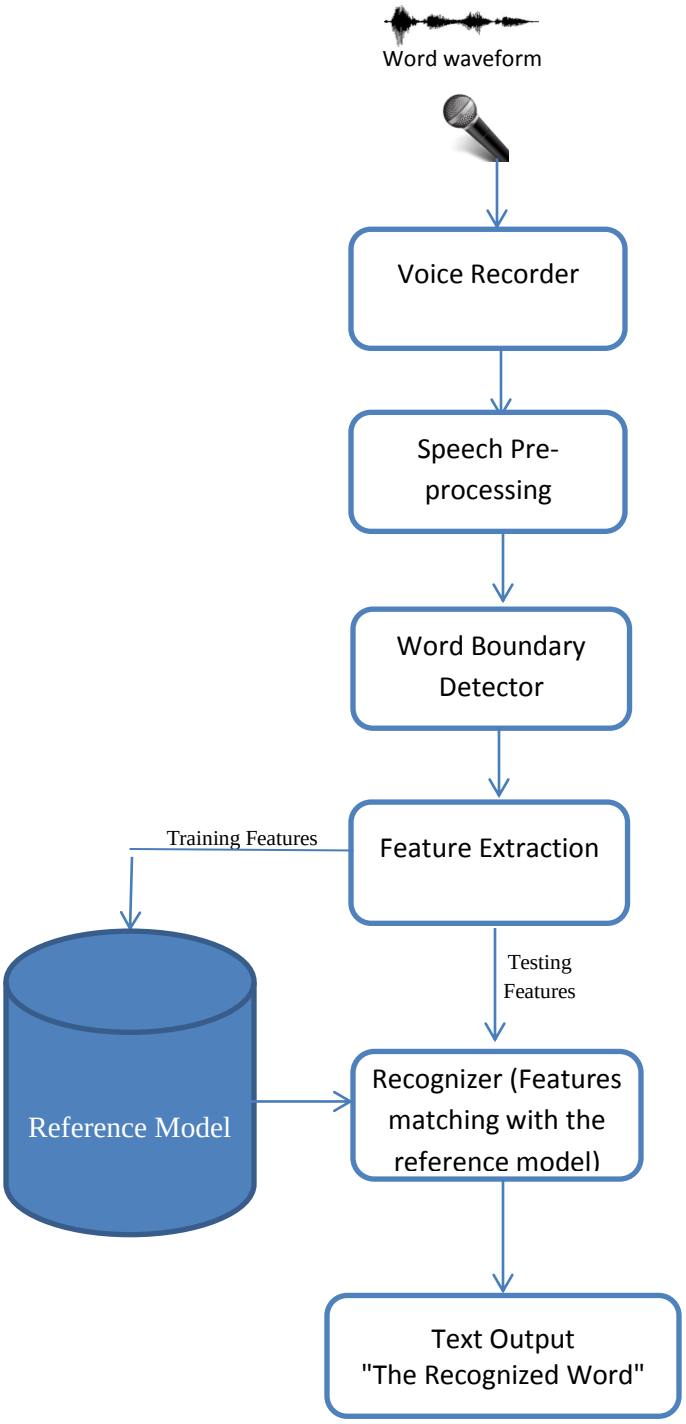

Figure 4.1 System block diagram

4.4 Voice Recording

Every user has to complete a voice recording session of 40 words repeated 8 times (5 of them are used in training and the remaining are used in the testing). Every word is recorded in 8000Hz, 16 bit, mono Wave file format.

In this step the continuous sound wave is converted to a sequence of discrete samples 8000 samples in each second since the sampling rate fs=8000 Hz.

We choose the value of fs=8000 Hz because for most phonemes in human speech, almost all of the energy is contained in the 5Hz-4 kHz range [42] (fmax=4000 Hz) and according to the Nyquist–Shannon sampling theorem [43] which states that perfect reconstruction of a signal is possible when the sampling frequency is greater than twice the maximum frequency of the signal being sampled, or equivalently:

$$Sampling\ Rate = 2 \times f_{max} = 4000 \times 2 = 8000Hz$$

Which is the sampling rate used by nearly all telephony systems.

4.5 Pre-Processing

Preprocessing is used before features extraction in order to reduce noise in speech signal and to enhance recognition accuracy.

4.5.1. DC Offset Removal

First the signal is pre-processed by removing the DC offset. The microphone with A/D converter may add a DC offset voltage to the output signal. Removing the DC offset is important in order to determine the boundary of words.

Before finding the boundaries of the word we remove DC offset of the speech signal by removing its mean.

$$speech = speech - mean(speech)$$

4.5.2. Normalization

Before extracting features of the speech signal we make normalization on speech signals to make the signals comparable regardless of differences in magnitude.

The normalization procedure is done by dividing the speech signal by maximum of absolute value of signal so that speech will be in the range from [-1,1].

$$speech = \frac{speech}{\max(|speech|)}$$

4.5.3. Discrete Wavelet Transform

Pre-processing of speech signals is considered a crucial step in the development of a robust and efficient speech or speaker recognition system. We applied discrete wavelet transform to the speech signal before extracting the features to improve the accuracy of the recognition and to make the system more robust to noise. We tested several wavelets families and levels: Haar, Daubechies 1, Daubechies 2, Daubechies 3, Daubechies 5, Daubechies 15, Coiflets, Symlets, Discrete Meyer; we find best result by using second level Daubechies wavelets.

The discrete wavelet transform divide the signal into approximation and detail coefficients, we take only the approximation coefficients vector as input for feature extraction stage.

4.6 End Point Detection (Word Boundary Detection)

We use end point detection to extract the word speech and remove the background noise and silence at the beginning and end of the word speech. End point detection improves performance of an ASR system in terms of accuracy and speed. Classification of speech into silence, voiced or unvoiced sounds provides a useful basis for subsequent processing.

- Silence, when no speech is produced.
- Unvoiced: Vocal cords are not vibrating, resulting in a periodic or aperiodic speech waveform.
- Voiced: Vocal cords are tensed and vibrating periodically, resulting in speech waveform that is quasi-periodic. Quasi-periodic means that the speech waveform

can be seen as periodic over a short-time period (5-100 ms) during which it is stationary.

The block diagram of the End Point Detection is shown in Figure 4.2:

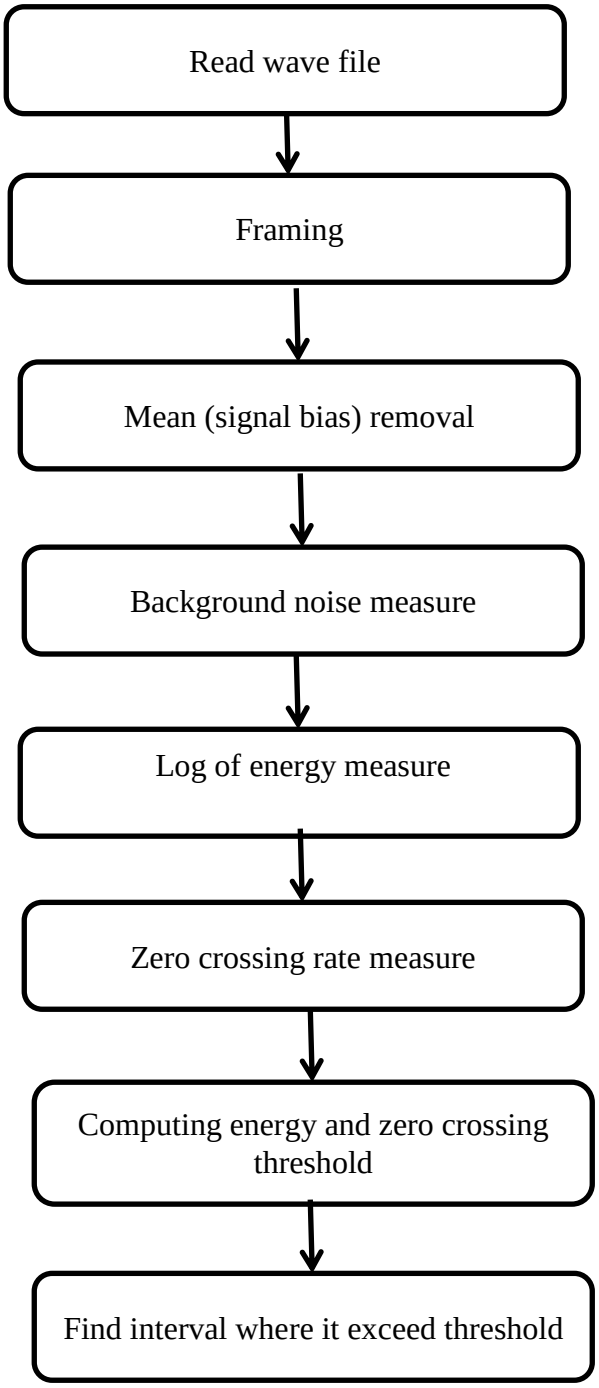

Figure 4.2 End point detection block diagram

The step-by-step computations of the End Point Detection algorithm are shown below:

i. Step1 : Framing

In order to have a stationary sound we need to divide the sound into small frames. Typical, ASR systems utilize a frame size between 10ms and 30ms with 50% frame overlap. In our system we used a frame size= 20 ms with 50% overlap.

Time frame=20ms

Fs=8000Hz

Number of samples in frame=Time frame×Fs=160 samples

Overlap=80 samples

ii. Step2 : Mean (signal bias) removal

In this step, we remove the mean (signal bias removal) for each frame to reduce the effect of noise in the frame.

$$frame = frame - mean(frame)$$

iii. Step3 : Background noise measure

Noise estimate is critical part and it is important for speech enhancement algorithms. If the noise estimate is too low then annoying residual noise will be available and if the noise estimate is too high then speech will get distorted and loss intelligibility. A common approach of noise estimation is to average the noise over nonspeech signals.

We recorded a silence signal. Then we estimate the noise in speech by computing the average energy and zero crossing rates of the silence signal frames.

iv. Step4 : Log of energy measure

Short-term energy is the principal and most natural feature that has been used. This is especially to distinguish between voiced sounds and unvoiced sounds or silence compared the performances of the following three short-times energy measurements in end point detection. It is observed that short-term energy is the most effective energy

parameter for this task. Voiced speech has most of its energy collected in the lower frequencies, whereas most energy of the unvoiced speech is found in the higher frequencies. The feeling of the sound intensity perceived by human ears is not linear but rather logarithmic. Thus, it is better to express the energy function in logarithmic form.

We use in the end point detection the zero crossing and the log of energy to find the boundary of the words since they are simple, fast and accurate.

- Logarithmic Short-Term Energy

Log energy Es is defined as:

$$Es = Log(e + \sum S(n)^2) \qquad (4.1)$$

Where S (n) is signal values in the frame and e is a small positive constant added to prevent the computing of log of zero. Generally speaking, Es for voiced data is much higher than the energy of silence. The energy of unvoiced data is usually lower than for voiced sounds but higher than for silence.

v. Step5 : Zero crossing rate measure

The number of zero-crossings refers to the number of times speech samples change sign in a given frame. It counts the number of zero crossing in the frame.
The zero crossing count is an indicator of the frequency at which the energy is concentrated in the signal spectrum. Voiced speech is produced as a result of excitation of the vocal tract by the periodic flow of air at the glottis and usually shows a low zero crossing count. Unvoiced speech is produced due to excitation of the vocal tract by the noise-like source at a point of constriction in the interior of the vocal tract and shows a high zero crossing count. The zero crossing count of silence is expected to be lower than for unvoiced speech, but quite comparable to that for voiced speech.

We use the following equation to compute the zero crossing rate.

$$Zcr(m) = \sum_{n=1}^{N} \frac{|sgn(S_m(n)) - sgn(S_m(n-1))|}{2} \quad (4.2)$$

Where:

$Zcr(m)$: *is the zero crossing rate in the frame* m

$$sgn(S_m(n)) \text{ is the sign function} = \begin{cases} +1, & S_m(n) \geq 0 \\ -1, & S_m(n) < 0 \end{cases} \quad (4.3)$$

$S_m(n)$: *is the speech signal in the sample number* n *in the frame* m

N: is the frame size

By combining the energy and the zero crossing rate, we can determine approximately the type of speech. Considering the voiced sound, the voiceless sound and the silent section, the short-time average scope of voiced sound is biggest and the short-time zero rate is lowest; the short-time average scope of voiceless sound comes between but the short-time zero rate is the highest; the short-time average scope of silent section is the lowest and the short-time zero rate comes between. This kind of comparison is relative but has no precise value relations.

vi. Step6 : Computing Energy Threshold

We need to find the threshold in order to decide where the utterance begins and where to end. The energy threshold can be described as,

$$T_E = \mu_E + \propto \times \sigma_E \quad (4.4)$$

Where: μ_E is the mean and σ_E is the standard deviation of the energy of the noise frames. The α term is constant that have to be fine tuned according to the characteristics of signal. We tested several values of α in the range from zero to one and we find that the best word boundary detection and system accuracy are with:

α=0.5

vii. Step7 : Computing Zero crossing Threshold

Zero-crossing rate threshold is defined as follows:

$$\mu_Z + \beta \times \sigma_Z \quad (4.5)$$

Where: μ_Z is the mean and σ_Z is the standard deviation of the zero crossing rates of the noise frames and β are parameters, which are obtained by experiments.

In our experiment, we find the good value for threshold factor is $\beta = 0.5$.

Also, according to many research the zero crossing rate of speech should be greater than 25 zero crossing per frame.

In our system we set the zero crossing thresholds to

$$T_Z = \max(\mu_Z + (\beta \times \sigma_Z\ ),\ 25) \quad (4.6)$$

viii. Step8 : Find interval where it exceeds threshold

By finding the energy and zero crossing rate threshold then the resultant threshold can be subtracted from the original speech. As a result pauses can be cleaned completely.

In this step, we test each frame by comparing its energy and zero crossing rates with the thresholds.

The pseudo code to find the start and end points of the word speech is shown below:

For each frame i in the speech signal

If $frame_{energy}(i) \geq \mathrm{T_E}$ *OR* $\mathrm{frame_{zerocrossing}(i)} \geq \mathrm{T_Z}$

Then mark this frame as the Start point of the possible word

Elseif Start is found AND 9 successive frames do not satisfy threshold criteria

Then End point is the first frame before the 9 successive frames

End

Calculate number of frames between Start and End points

If it is greater than 25 frames (0.5 second).

Then a word is detected

Else we disregard it and we repeat the procedure to find other possible words.

End

End

The detected word is saved to be used for the feature extraction phase.

Figure 4.3 and Figure 4.4 are plots of the signal of a spoken word "أمام" before and after applying end point detection algorithm.

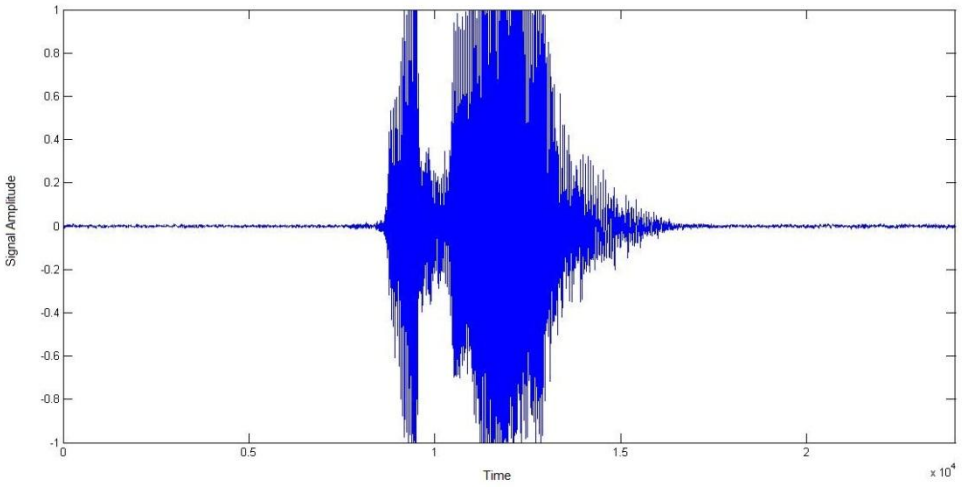

Figure 4.3 The waveform of the word امام before applying the end point detection

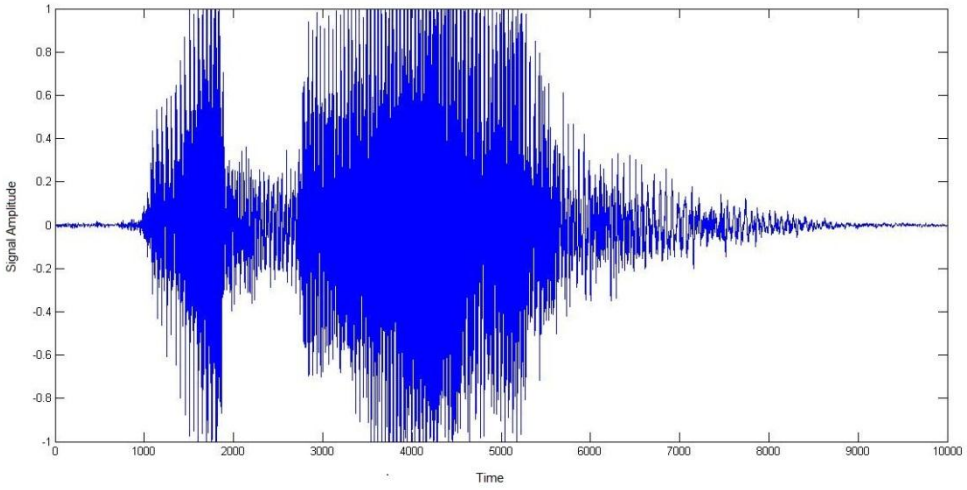

Figure 4.4 The waveform of the word امام after applying the end point detection

In Figure 4.3, the signal length is 3 seconds or 24000 samples

Number of samples =Time $\times$*Fs*=3 *seconds* $\times$*8000Hz=24000 samples*

In Figure 4.4, the signal length of the word أمام after applying the end point detection algorithm reduced to 10000 samples or 1.25 seconds (10000samples/8000 Hz). Where silence at the beginning and end of the word is removed. This will reduce the computation time of the feature extraction and recognition in the system to about 58% ((3sec-1.25sec)/3sec).

4.7 Feature Extraction

A key assumption made in the design of most speech recognition systems is that the segment of a speech signal can be considered as stationary over an interval of few milliseconds. Therefore the speech signal can be divided into blocks which are usually called frames. The spacing between the beginnings of two consecutive frames is in the order of 10 msecs, and the size of a frame is about 25 msecs. That is, the frames are overlapping to provide longer analysis windows. Within each of these frames, some feature parameters characterizing the speech signal are extracted. These feature parameters are then used in the training and also in the recognition stage [44].

In this thesis, a combination of several famous features (MFCC, LPC, Formants) has been used to improve the accuracy of the system.

4.7.1. Mel-Frequency Cepstral Coefficients (MFCC)

MFCC is one of the best known and most commonly used features for speech recognition. MFCC is applied with frame length of 256 samples which is equal to 32 msecs, and with frame overlap of length 80 samples which is equal to 10 msecs (1 second of recording is equal to 8000 data points). These values are found to result in most accurate recognition after several experiments. The output of MFCC is a 22 dimensional matrix where the number of rows is the number of frames of the speech signal. The dimension is fixed to 22, which represents the length of the vocal tract of humans. The Block diagram of MFCC is shown in the Figure 4.5:

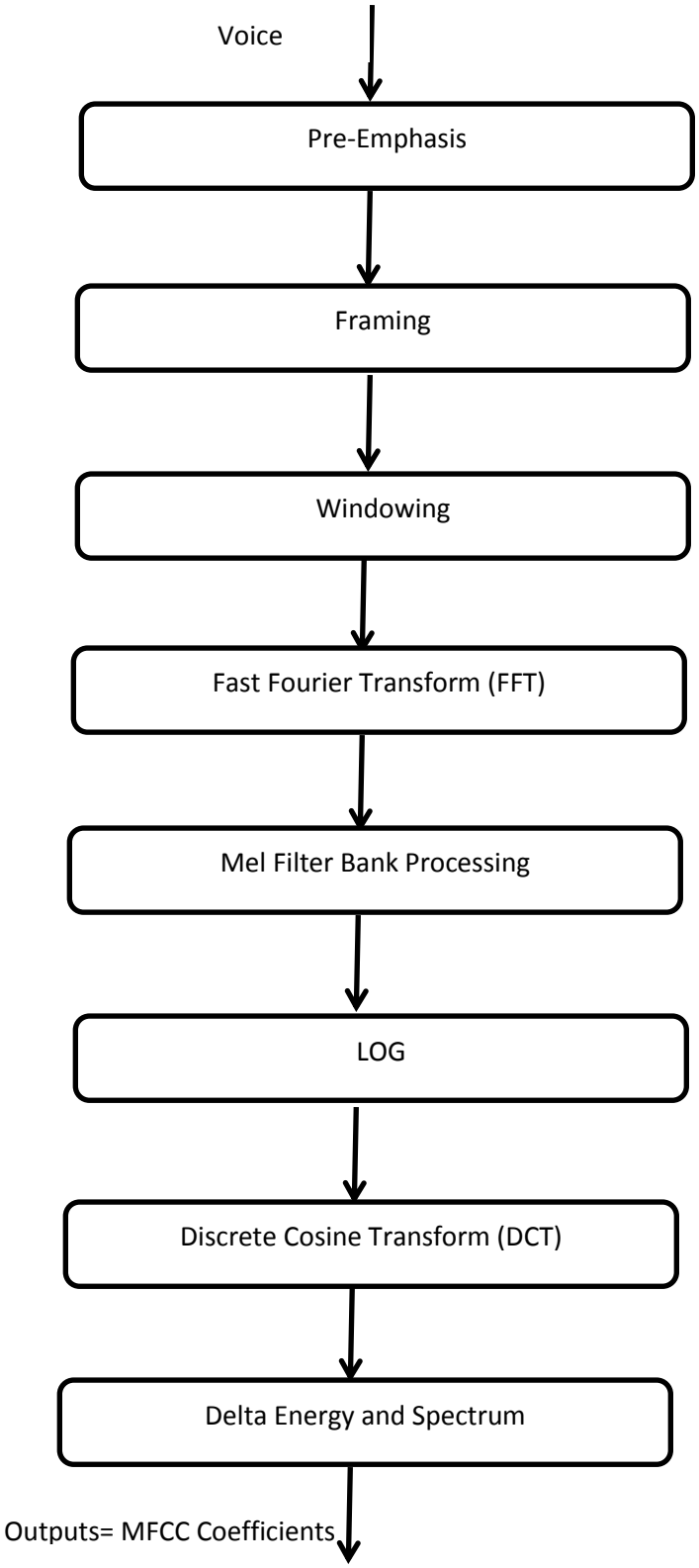

Figure 4.5 MFCC block diagram

MFCC takes human perception sensitivity with respect to frequencies into consideration, and therefore is best known and most commonly used feature in speech recognition. The step-by-step computations of MFCC are shown below:

a. Step1 : Framing

In this step, the signal is divided into small blocks of samples called frame. The length of frame must be in 10-40ms short time duration to ensure stationary of signal. In this work, the frame length is set to 32ms (or N=256 samples in the frame).

Where the block size (N) must be a power of 2 to speed up the computation.

b. Step2 : Windowing

Windowing is a point-wise multiplication between the framed signal and the window function. In order to eliminate the errors due to computing the Fast Fourier Transform (FFT) on a block of data (Frame) which is not periodic [45], we need first to apply a window function to smooth the signal in the data block and to eliminate unwanted signal like noise and interference joined with the signal. A window is shaped so that it is exactly zero at the beginning and end of the data block and has some special shape in between. This function is then multiplied with the time data block forcing the signal to be periodic. In our system we choose the hamming window since it is good in improving the frequency resolution. That is, they make it easier to detect the exact frequency of a peak in the spectrum and minimize the spectral distortion and the signal discontinuities. Hamming window function shape is shown in Figure 4.6.

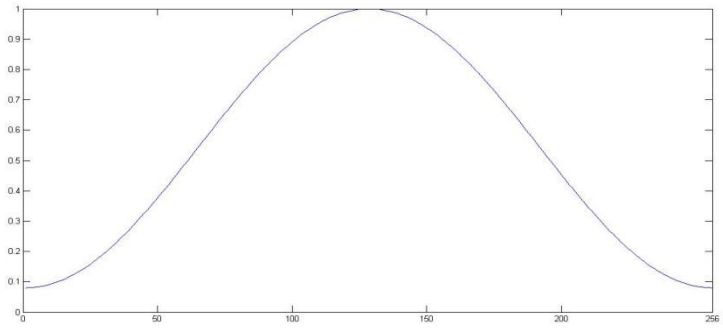

Figure 4.6 Hamming window for FFT block of 256 samples

One of the disadvantages of windowing functions is that the beginning and end of the signal is attenuated in the calculation of the spectrum. Overlap processing recover a portion of each previous frame that otherwise is lost due to the effect of the windowing function. So, in our system we used an overlapped hamming window in computing the voice spectrum. We set the time shift of the overlapped windows as follow:

Overlap =10 ms =0.01 seconds

Data shift =time shift × fs= 0.01 sec × 8000 Hz =80 samples

c. Step 3 : Fast Fourier Transform (FFT)

The purpose of FFT is to convert the signal from time domain to frequency domain to get the frequency content of speech signal in current frame.

$$Y(f) = FFT(\, x_m(t) \times \, H(t) \,)$$

Where:

Y(f) : is the Fourier Transform of x

H(t) : is the hamming windows over a frame of 256 samples

$x_m(t)$: is the m frame in the speech signal of length 256 samples and 80 samples overlap.

d. Step 4 : Mel Filter Bank Processing

Mel Filter Banks are a set of triangular band pass filters used to approximate the frequency resolution of the human ear, which resolves frequencies non-linearly across the audio spectrum. The information carried in low frequency components of the speech signal is more important than the high frequency components. In order to concentrate on the low frequency components, Mel scaling is applied. The Mel frequency scale is linear up to 1000 Hz and logarithmic thereafter.

To convert from frequency scale to Mel scale we use the following equation [46]:

$$Mel(f) = 2595\ log_{10}(1 + \frac{f}{700}) \quad (4.7)$$

To convert from Mel scale to frequency scale we use the following equation:

$$\text{Freq(m)} = 700 \times (10^{\frac{m}{2595}} - 1) \quad (4.8)$$

The literature indicates that 12 Mel filter is typically sufficient for speaker recognition and can vary for different implementations up to 40 Mel filters.

In our system we use 22 triangular Mel filters,which represents the length of the vocal tract of humans, spread over the whole frequency range from zero upto the Nyquist frequency fmax =fs/2=4000 Hz.

To get the maximum Mel frequency melmax we use eq. (4.7)

melmax = 2595 ×log10(1 + fmax /700)=2595 × log10(1 + 4000 /700)=2146

The center frequencies of the 22 triangular filters "melcenters" will be linearly spaced along the Mel scale.

melcenters = (1:nofChannels) × melmax / (nofChannels + 1)=(1:22) * 2146/(22+1)=

[93.3 186.6 279.9 373.2 466.5 559.8 653.1 746.4 839.7 933 1026.3 1119.6 1212.9 1306.2 1399.5 1492.8 1586.1 1679.4 1772.7 1866 1959.3 2052.6]

To obtain the center frequencies of the filter on the frequency scale fcenters, we use eq. (4.8) and we add 1 and 4000 to the vector.

fcenters = [1 60.4 126 197.3 274.8 358.9 450.3 549.6 657.5 774.6 901.9 1040.2 1190.4 1353.5 1530.7 1723.3 1932.4 2159.6 2406.5 2674.6 2965.8 3282.2 3625.9 4000]

Which are nonlinearly spaced.

To find to which fft block this fcenters belong:

We have fs=8000 is divided into equal spaced N=256 fft blocks or the nyquist frequency fmax =4000 is divided into equal spaced Nyquist frequency index Nmax = N/2=128 fft blocks.

Then the frequency resolution: df = fs/N=8000/256=31.25

FFT indices: f = round (fcenters ./ df) = round (fcenters ./ 31.25)=

[1 2 4 6 9 11 14 18 21 25 29 33 38 43 49 55 62 69 77 86 95 105 116 128]

Then we use eq. (4.9) to find the coefficient of the triangular filters.

$$w(c,i)_{c=1:22} = \begin{cases} 0 & ,\text{for } f(i) \le f(c-1) \\ \frac{f(i)-f(c-1)}{f(c)-f(c-1)} & ,\text{for } f(c-1) \le f(i) \le f(c) \\ \frac{f(c+1)-f(i)}{f(c+1)-f(c)} & ,\text{for } f(c) \le f(i) \le f(c+1) \\ 0 & ,\text{for } f(i) \ge f(c+1) \end{cases} \quad (4.9)$$

Then we normalize the coefficient to have a unit area of each triangle filter

$$W(c,:) = w(c,:)/\,\text{sum}(w(c,:))$$

The plot of these triangular filters is shown in Figure 4.7.

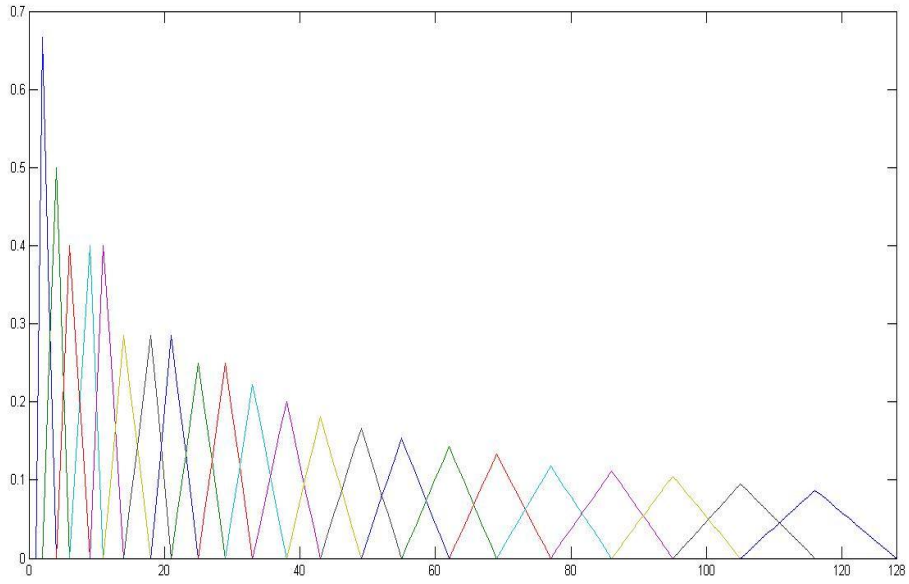

Figure 4.7 Mel-scale filter bank

The filters have not equal gain because of their varying frequency widths (in linear scale).

e. Step 5 : Calculating the logarithm

In this step, we multiply this filters with power spectrum obtained in step 3 and we normalize it and we calculate the logarithm of each Mel power spectrum coefficient.

f. Step 6 : Discrete Cosine Transform (DCT)

In this step, log Mel spectrum is converted back to time domain using DCT. The result of conversion is called Mel Frequency Cepstrum Coefficients (MFCCs).

4.7.2. Formant and LPC Features

We use Formant_tracker based on LPC Analysis to estimate the first 3 formants (F1, F2 and F3) of each frame in the speech signal.

To compute the LPC coefficient, we use Levinson-Durbin Algorithm.

a. LPC Coefficient computation

- Step1 Framing

First the signal is divided into frames of length 256 samples with 50 % overlapping

- Step2: Choose p the number of LPC coefficients

For accurate vocal tract model: The order of the LPC should be greater than sample rate/1000 + 2.

With sampling rate 8 kHz, the typical order of the LPC should be greater than 8000/1000+2=10 (we select p=12).

- Step 3: Auto-Correlation

Linear predictive coding (LPC) is a popular technique for speech compression. Since speech signals are not stationary, we are typically interested in the similarities in signals only over a short time duration (frame) and for only a few time delays $i=\{0,1,\ldots,P\}$. So, we want to remove redundancy in each frame to reduce the number of features. Autocorrelation describes the redundancy in the speech frame x(n).

For each speech frame we calculate the 12 auto-correlation coefficients. The auto-correlation formula is shown below in Eq. (4.10).

$$R(i) = \frac{1}{N}\sum_{n=1}^{N-1} x(n).x(n-i) \quad for\ i = 0\ to\ p \tag{4.10}$$

Where:

N: is the frame length N-1=511

p : is the number of LP coeffcients (LPC order) p=12

And the $\frac{1}{N}$: is a normalizing factor.

The auto-correlation coefficient, R0, is the energy of the window frame.

- Step4: Levinson-Durbin algorithm

The 13 auto-correlation coefficients R (i) are combined with Levinson-Durbin algorithm to find the 12 LPC coefficients.

Each window frame is modeled as an IIR filter with the following transfer function shown below in Eq. (4.11).

$$H(z) = \frac{G}{(1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{12} z^{-12})} \tag{4.11}$$

The 12 LPC coefficients (a_1, a_2, …. , a_{12}) is the solution to the auto-correlation of Eq. (4.12).

$$\begin{bmatrix} R(0) & R(1) & R(11) \\ R(1) & R(0) & R(10) \\ \cdots & \cdots & \cdots \\ R(11) & R(10) & R(0) \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_{12} \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \cdots \\ R(12) \end{bmatrix} \tag{4.12}$$

The solution is calculated using Levinson-Durbin recursive algorithm as follow:

Algoritm 4.1: Levinson-Durbin recursive

Purpose: LPC Coefficient computation

Input:

R(i): autocorrelation coefficients

a(i) : filter coefficients

K : reflection coefficients

E: prediction gain

Output: The 12 LPC coefficients, a(1), a(2), a(3), …., a(12).

Initialization:

a(0) = 1 and E(0) = R(0)

k(1)=R(1)/R(0)

alpha(1,1)=k(1)

E(1)=(1-k(1)^2)×E(0)

Procedure:

$for \;\; 2 \leq i \leq p$=12

$k(i) = \frac{1}{E(i-1)} \; (R(i) - \sum_{j=1}^{i-1} \propto (j, i-1) \, R(|i-j|)$

$\propto (i,i) = k(i)$

$for \quad 1 \leq j \leq i-1$

$\propto (j,i) = \propto (j, i-1) - k(i) \propto (i-j, i-1)$

end

$E(i) = (1 - k(i)^2) E(i-1)$

end

the final solution of the a's coefficients is given by

$for \; 1 \leq j \leq p = 12$

$a(j) = \propto (j,p)$

end

The LPC Gain Coefficient G is given by

$$G^2 = R(0) - \sum_{k=1}^{p} a(k) \, R(k)$$

where a(1), a(2), a(3), …., a(12) are the 12 LPC coefficients.

b. Formants

Given a spectral representation S(z) via the LPC coefficients, one could directly locate formants by solving directly for the roots of the denominator polynomial in S(z). Each complex-conjugate pair of roots would correspond to a formant. For each LPC_Frame, The formant frequencies are obtained by finding the angle of the roots of the LPC prediction polynomial.

rts = roots(LPC Coefficients).

Because the LPC coefficients are real-valued, the roots occur in complex conjugate pairs. We retain only the roots with positive imaginary part and determine the angles corresponding to the roots.

rts = rts(imag(rts)>=0).

angles = arctan(imag(rts),real(rts)).

Then we Convert the angular frequencies in radians/sample represented by the angles to hertz and calculate the bandwidths of the formants.

$$\text{Formants frequencies} = \frac{angles \times F_s}{2\pi}$$

We sort these Formants frequencies then we take only the first 3 Formants, since they are the most important in determining the uttered word.

c. Itakura Distance (comparing two sets of LPC coefficients)

Given two vectors of LPC coefficients, it is often necessary to compute the "distance" between two LPC vectors in pattern recognition application such as speech recognition. The euclidean and manhattan distance measures are not appropriate for comparing two vectors of LPC coefficients since the coefficients are not independent. The most useful distance measures for LPC coefficients are Itakura distance. The LPC coefficients aim to minimize the residual energy between the true magnitude spectrum of the speech frame

and the LPC model spectrum. This suggests that one may compute the "distance" between two LPC vectors by comparing their residual energies between each of their reconstructed spectra and "true" spectrum.

Let a and $\hat{a}$ be the pth-order LPC coefficients computed from two (windowed) speech frames x(n) and x($\hat{n}$) respectively. It is known that the prediction error (residual energy) by linear predictive analysis can be written in the form:

$$E^{(p)} = a^T R_x a \tag{4.13}$$

Where Rx is the Toeplitz matrix calculated from the autocorrelation sequences of the signal x(n). Thus, a reasonable measure of spectral distance between two frames of speech represented by a and $\hat{a}$, and augmented matrices R and $\hat{R}$ is

$$D(a, \hat{a}) = \frac{\hat{a}^T R_x \hat{a}}{a^T R_x a} \tag{4.14}$$

Itakura distance is defined as

$$D_I(a, \hat{a}) = \log \frac{a^T R_x a}{\hat{a}^T R_{\hat{x}} \hat{a}} \tag{4.15}$$

4.8 Training Stage

In the training stage we create the reference model for the training speech signals. This reference contains the LPC, MFCC and Formants features and their gaussian mixture models.

4.8.1. Training with Gaussian Mixture Model

To create the reference model, we use Gaussian mixture model to fit the extracted features of the training data. Gaussian Mixture Models form clusters by representing the probability density function of observed variables as a mixture of multivariate normal densities. Mixture models of the gmdistribution class use expectation maximization (EM) algorithm to fit data, which assigns posterior probabilities to each component density with respect to each observation. Clusters are assigned by selecting the component that maximizes the posterior probability. The posterior probabilities for each point indicate that each data point has some probability of belonging to each cluster. Gaussian mixture modeling uses an iterative algorithm that converges to a local optimum.

To find the gaussian mixture model for each word that fit the training data and estimate its parameters, we use the matlab command:

obj = gmdistribution.fit(X,k,param1,val1,param2,val2,..,'Options',options)

Where:

k: is the number of Gaussian mixture components. We set k=5 since each word is repeated five times and the word model will contains five normal distribution.

X: is the word features in n-by-d matrix, where n is the number of observations (number of frames) and d is the dimension of the data (number of features in each frame d=22).

The algorithm starts by an initial point randomly, in order to have same result each time we run the code we have to make this initial point the same when the algorithm start . We need to reset MATLAB's random number generator, and the simplest way to that is

reset(RandStream.getDefaultStream)

The parameter/values:

'Replicates': A positive integer giving the number of times to repeat the EM algorithm, each time with a new set of parameters. The solution with the largest likelihood is returned. In our system we set (Replicates=3) which gives best accuracy.

'CovType': we restricted the covariance matrices to be 'diagonal' instead of full covariance matrices in order to decrease the number of model parameters

'Options': Options structure for the iterative EM algorithm, as created by statset command. We used the parameters:

'Display' with value of 'off' which is the default (Displays no information.).

and 'MaxIter' with value of 500 (Maximum number of iterations allowed).

4.9 Recognition Stage (Test Phase)

In the recognition stage a combination of recognition methods are used.

4.9.1. Euclidean Distances

We use a Pairwise Euclidean distances between columns of MFCC test features matrix with each MFCC training matrices in the reference models.

let:

x: MFCC test features

y :MFCC training features

First we calculate the Euclidean distance D between each column in x with each column in y.

$$D = \sqrt{\sum (x - y)^2}$$

Then we find the minimum m value of each row in D

The distance d between x and y will be the average of m

$$d=Average(m)$$

We repeat the above procedure to find the distance d between x and each training vector.

The training vector that has the smallest distance d to the test vector x is the recognized word.

4.9.2. Dynamic Time Warping(DTW)

Dynamic Time Warping (DTW) is a technique that finds the optimal alignment between two time series if one time series may be "warped" non-linearly by stretching or shrinking it along its time axis. This warping between two time series can then be used to find corresponding regions between the two time series or to determine the similarity between the two time series.

Speech is a time-dependent process. Hence the utterances of the same word will have different durations, and utterances of the same word with the same duration will differ in the middle, due to different parts of the words being spoken at different rates. To obtain a global distance between two speech patterns (represented as a sequence of vectors) a time alignment must be performed. DTW resolves this problem by aligning the words properly and calculating the minimum distance between them. The local distance measure is the distance between features at a pair of frames while the global distance from beginning of utterance until last pair of frames reflects the similarity between two vectors. We used dynamic time warping to classify the MFCC and Formants features. Since these features have not the same dimension. The algorithm of DTW is as follow:

Algoritm 4.2: Dynamic Time Warping

Purpose: Global distance between testing and training features

Input:

X: test features Formants

Y: training features Formants

Size(X) =[r, n]

Size(Y) =[r, m]

X and Y have same number of rows but different number of column $m \neq n$

D: Global distance, an n × m matrix.

Output: dist=D (n, m) the global distance.

Initialization:

Set all elements values in D to infinity.

Set the start element in D to zero, D (1, 1) =0.

Procedure:

for i=1:n

for j=1:m

$d = \sqrt{\sum_{k=1}^{r}(X(k,i) - Y(k,j))^2}$ where d: is the local distance (Euclidean distance between the two feature points and r is number of rows)

D(i,j)=d+ minimum of (D(i-1 ,j), // insertion

D(i , j-1), // deletion

D(i-1, j-1)) // match

end

end

Comparing the test features with each of the training features the one that have the smalest value of "dist " is considered the recognized word.

4.9.3. Gaussian Mixture Model GMM Recognizer

During the testing stage, we extract the MFCC vectors from the test speech and compare it with estimating GMM model of each word and use a probabilistic measure to determine the source word with maximum a posteriori probability (maximizing a log-

likelihood value). The log-likelihood value is computed using the posterior function in matlab: *[P,nlogl] = posterior(obj,X)*

nlogl: is the negative log-likelihood function. It is the negative sum of logs. Typically, we search for minimum rather than maximum where the minimum value is used to classify the input speech from spoken words.

4.10 Speaker Recognition

In order to accelerate the system and to reduce the execution time, we first make the system to identify the speaker and load only its reference model. Each speaker will read the sentence "بسم الله الرحمن الرحيم". Then we Calculate MFCC coefficients for training set.
To reduce the speaker identification time, we use Vector Quantization (K-means) which map the large MFCC features size to to a predefined number of clusters defined by their centroids. As the number of centroids increases, the identification rate of the system increases, but the computational time will also increase. However, increasing the number of centroids, greater than 64, did not have any further impact [48].
For clustering we use 64 centroid K-means clustering where squared Euclidean distance is used and each centroid is the mean of the points in that cluster. We use the matlab command

[IDX,C,sumd] = kmeans(X,k)

Where:
X: is the mfcc feature of the speech signals.
k: is the number of clusters we set k=64.
IDX: are the cluster indices of each point.
C: centroid locations matrix.
sumd: returns the within-cluster sums of point-to-centroid distances.

In the recognition phase we extract the MFCC features of the test speech signal and we use K-means clustering to find centoids matrices "C" for the test features. Then we compute the average distances between the features of the unknown voice test_C with all the codebooks in database train_C, and find the lowest distortion and identify the person with the lowest distance.

Chapter 5
Experimentation and Results

5.1 System Datasets and Parameters

The dataset used in the test phase consist of 600 samples recorded with same training speakers in a clean environment (5 Speakers * 40 Words * 3 Repititions = 600 samples). Where sounds are recorded using laptop microphone with sampling frequency of 8 kHz, 16-bit PCM Mono WAV format.

The speech recognition system is implemented and tested using Matlab R2010a and the Laptop used in recording of sound and computation of the system is:

HP G62-a10se Laptop, Core I3/2.26Ghz processor, 2 GB RAM, built-in Realtek High Definition Audio Sound and Windows7 Ultimate Operating System.

Table 5-1 System parameters

System Parameters	Values
Training Dataset	1000 Samples
Testing Dataset	600 Samples
Computer	HP G62-a10se Laptop
Processor	core I3/2.26Ghz
RAM	2 GB
Audio Sound	built-in Realtek High Definition Sound Card
Software	Matlab R2010a
Operating System	Windows7 Ultimate

5.2 System Graphic User Interface (GUI)

We designed the system in a graphic user interface GUI in Matlab to make it simple to use. First when we run the software a dialog box will appear asking to read the sentence"بسم الله الرحمن الرحيم", as shown in Figure 5.1.

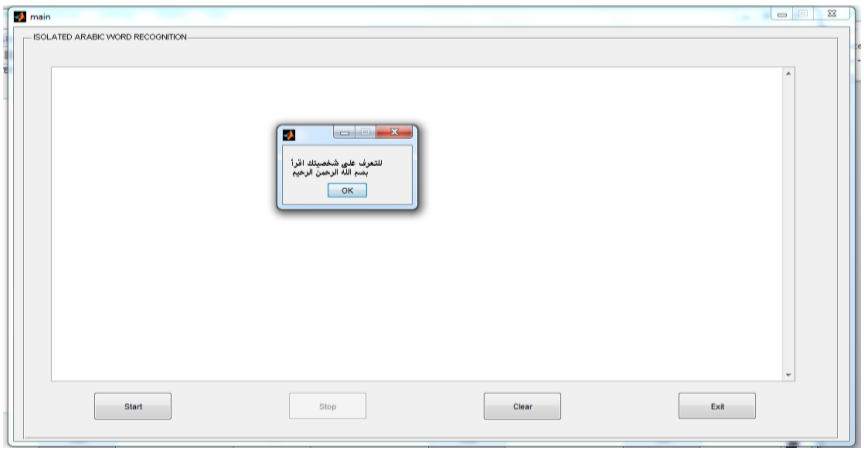

Figure 5.1 System GUI speaker identification dialog box

Then the system will recognize the user and will display his name in the top of the canvas as shown in Figure 5.2. Then it will load only the reference model obtained in training only for that user and discard other user models this will reduce the time of speech recognition by comparing his test speech with only his training data.

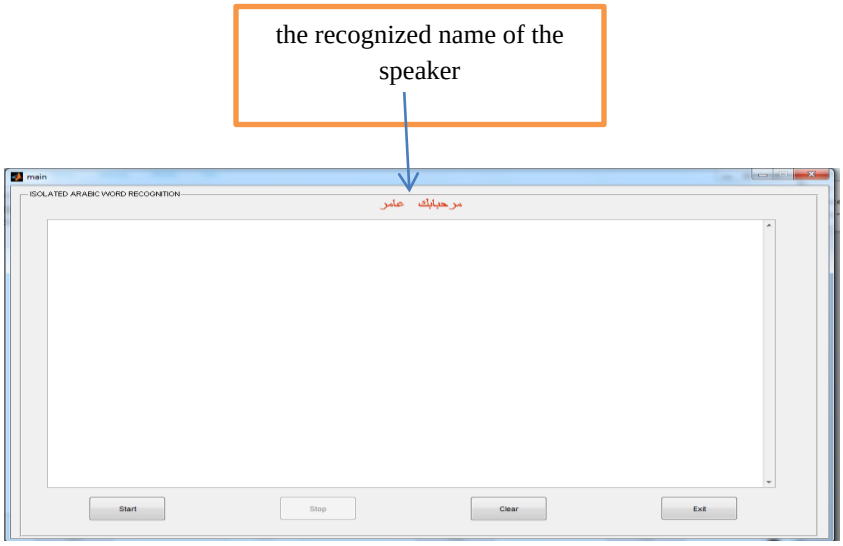

Figure 5.2 The recognized speaker name displayed on top of canvas

We have in the GUI 4 buttons:

- Start button : when pressed the system will start recording the sound for 2 minutes then recognizes the word
- Stop button : used to stop every thing and remove any occuring errors
- Clear button :used to clear the workspace and command windows in matlab and the textbox of the canvas
- Exit button : used to close the program and exit

The recognized word will appear in the textbox and each time you press the start button and read new word, it will be displayed next to it. Figure 5.3 shows an example of reading 3 words.

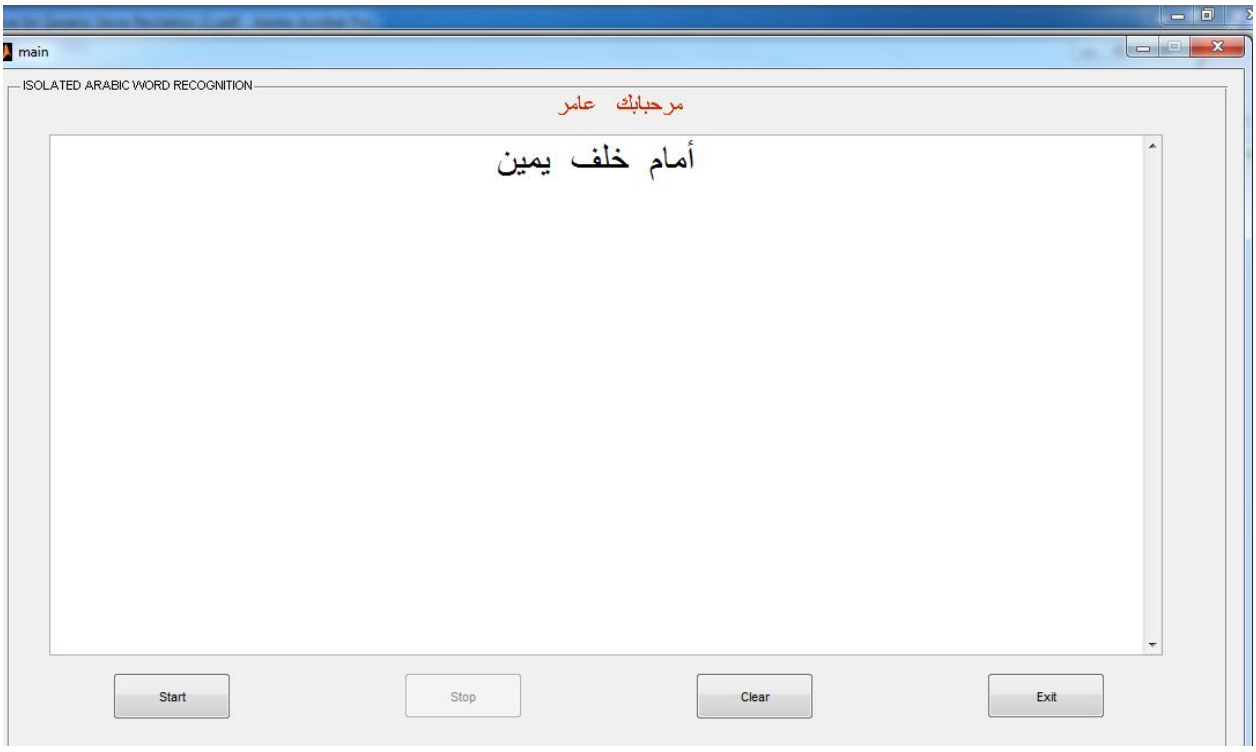

Figure 5.3 Example of reading 3 words

5.3 Recognition Methods Experiments and Results

We have performed different experiments and obtained the results shown in the tables below.

1. using MFCC features and Pairwise Euclidean distances Classification
2. using FORMANTS features and Dynamic time warping (DTW) Classification
3. using MFCC features and Gaussian mixture model (GMM) Classification
4. using MFCC features and Dynamic time warping (DTW) Classification
5. using LPC coefficients features and Itakura distance Classification
6. Combination of the methods outputs using a voting system.

In order to evaluate the recognition rate for each method, we calculate the method accuracy for each speaker using its 120 test data (40 words repeated 3 times). Then the overall accuracy of the method is the average accuracy of the 5 speakers.

To visualize the performance of the mehods, we used a confusion matrix which is a simple matching matrix used to display the classification results. The confusion matrix is defined by labeling the desired classification on the rows and the predicted classifications on the columns. Since we want the predicted classification to be the same as the desired classification, the ideal situation is to have the values of the matrix diagonal equal to 15. Which is the total number of test data of each word (5 speaker * 3 repitition of each word).

In order to find the accuracy of each word in the table we divide the diagonal value in the table of this word over 15. For example the word "خلف"has a diagonal value in the first table equal to 14 then 14/15*100%= 94% which is the accuracy of this word in the first table.

In Table 5.2, we notice that 11 words (.....، "يمين" ،"أمام") from the 40 words get 100% accuracy using the MFCC feature with Euclidean classification. Whereas, The worst case was with the words ("احمل" ، "أغلق") with an accuracy of 60%. The performance of this method is good. It is 85.23%.

Table 5-2 Confusion matrix of the system when using MFCC features and pairwise Euclidean classification

	أمام	خلف	يمين	يسار	أعلى	أسفل	تحرك	قف	أسرع	تمهل	افتح	أغلق	انزل	اصعد	اقرأ	اكتب	تكلم	اسكت	أجب	فوق	ابدأ	توقف	أكمل	امسح	احمل	انظر	انطلق	أعد	نعم	لا	صفر	واحد	اثنين	ثلاثة	أربعة	خمسة	ستة	سبعة	ثمانية	تسعة	Accuracy%
أمام	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
خلف	1	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
يمين	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
يسار	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	94%
أعلى	0	0	0	0	**12**	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	80%
أسفل	1	0	0	0	0	**11**	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
تحرك	0	0	0	1	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	87%
قف	0	0	0	0	0	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
أسرع	0	0	0	0	1	1	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	80%
تمهل	1	0	0	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
افتح	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	74%
أغلق	0	0	0	0	1	0	0	0	1	0	0	**9**	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	60%
انزل	0	0	1	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
اصعد	0	0	0	0	0	0	0	0	1	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	74%
اقرأ	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	80%
أكتب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
تكلم	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	100%
اسكت	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	80%
أجب	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	**10**	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	67%
فوق	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
ابدأ	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	74%
توقف	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	74%
أكمل	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
امسح	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	87%
احمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	0	**9**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60%
أنظر	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
انطلق	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
أعد	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	100%
نعم	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	100%
لا	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**10**	0	0	0	0	1	0	0	2	0	0	67%
صفر	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	1	0	0	0	80%
واحد	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	**11**	0	0	0	0	1	0	0	74%
اثنين	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**12**	0	0	0	0	1	0	80%
ثلاثة	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**12**	0	0	0	0	0	0	80%
أربعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	100%
خمسة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	100%
ستة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	100%
سبعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	100%
ثمانية	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	100%
تسعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	100%
TOTAL ACCURACY % = **85.23 %**																																									

As shown in Table 5.3, the total accuracy of this Formant based DTW classification method is very low only 57%. We may conclude that Formant based recognition system should not taken alone and should be merged with other features to have better accuracy. The best accuracy is with the word "سبعة"with 94% and the worst one is the words ("اقرأ" ،"ابدأ") with only 27%.

Table 5-3 Confusion matrix of the system when using Formants features and DTW classification

	أمام	خلف	يمين	يسار	أعلى	أسفل	تحرك	قف	أسرع	تمهل	افتح	أغلق	انزل	اصعد	اقرأ	اكتب	تكلم	اسكت	أجب	فوق	ابدأ	توقف	أكمل	امسح	احمل	انظر	انطلق	أعد	نعم	لا	صفر	واحد	اثنين	ثلاثة	أربعة	خمسة	ستة	سبعة	ثمانية	تسعة	Accuracy%
أمام	**9**	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	60%
خلف	0	**8**	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	54%
يمين	0	0	**9**	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60%
يسار	0	0	0	**7**	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	46%
أعلى	0	0	0	0	**7**	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	46%
أسفل	1	0	0	0	0	**6**	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	40%
تحرك	0	0	0	1	0	1	**8**	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	54%
قف	0	0	0	0	0	0	0	**10**	0	0	0	1	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	67%
أسرع	0	0	0	0	0	1	0	0	**10**	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	67%
تمهل	0	0	0	0	0	2	1	1	0	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	67%
افتح	0	0	0	1	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	80%
أغلق	0	0	0	0	0	0	0	0	0	0	0	**9**	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	60%
انزل	0	0	1	0	0	0	0	0	0	0	0	0	**9**	0	0	1	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	60%
اصعد	0	0	0	0	0	0	0	0	0	0	0	0	0	**7**	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	3	46%
اقرأ	0	0	0	1	0	0	0	0	0	0	3	0	0	0	**4**	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	2	0	0	1	0	1	27%
أكتب	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	**8**	0	1	1	0	0	0	0	0	0	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0	54%
تكلم	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	**7**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	46%
اسكت	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	**7**	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	46%
أجب	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**9**	0	0	0	1	0	1	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	60%
فوق	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	**8**	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	54%
ابدأ	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**4**	1	0	1	0	0	3	0	0	0	3	0	0	1	0	0	1	0	0	0	27%
توقف	0	0	0	0	0	0	0	2	1	0	0	0	0	0	2	0	0	0	1	0	0	**5**	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	2	0	0	34%
أكمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	**10**	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	67%
امسح	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	74%
احمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	3	0	**7**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46%
أنظر	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	**9**	1	0	0	0	0	0	0	0	0	1	0	0	0	0	60%
انطلق	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	0	0	0	1	0	0	2	0	0	0	74%
أعد	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	**6**	0	0	0	2'	1	0	0	0	0	0	1	1	40%
نعم	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	**8**	0	1	0	0	0	0	1	0	1	0	1	54%
لا	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**8**	0	0	0	0	1	0	0	2	0	2	54%
صفر	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**8**	0	0	1	0	1	0	0	1	0	54%
واحد	2	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	**8**	0	0	0	0	0	1	1	0	54%
اثنين	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	**9**	0	0	0	1	0	0	1	60%
ثلاثة	0	0	0	0	0	0	0	1	0	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	**8**	0	0	0	0	1	0	54%
أربعة	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**10**	0	0	3	0	0	67%
خمسة	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	**10**	0	1	1	0	67%
ستة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	1	0	0	0	0	0	0	**8**	1	0	2	54%
سبعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	0	1	94%
ثمانية	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	8	0	54%
تسعة	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	**13**	87%
TOTAL ACCURACY % = **57 %**																																									

In Table 5.4, The Average accuracy of the system based on MFCC features and Gaussian Mixture Model is good equal to 87%. Using this method, 9 words have best accuracy of 100% and the worst case is with the words (" ابدأ "، "نعم") with accuracy of 67%.

Table 5-4 Confusion matrix of the system when using MFCC features and Gaussian Mixture Model (GMM) classification

	أمام	خلف	يمين	يسار	أعلى	أسفل	تحرك	قف	أسرع	تمهل	افتح	أغلق	انزل	اصعد	اقرأ	اكتب	تكلم	اسكت	أجب	فوق	ابدأ	توقف	أكمل	امسح	احمل	انظر	انطلق	أعد	نعم	لا	صفر	واحد	اثنين	ثلاثة	أربعة	خمسة	ستة	سبعة	ثمانية	تسعة	Accuracy%
أمام	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	94%
خلف	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
يمين	0	0	**13**	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
يسار	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	87%
أعلى	0	0	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
أسفل	0	0	0	0	1	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
تحرك	0	0	0	1	0	1	**12**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
قف	0	0	0	0	0	0	0	**13**	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
أسرع	0	0	0	0	0	2	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	80%
تمهل	0	1	0	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	80%
افتح	0	0	0	0	0	0	0	0	0	0	**14**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
أغلق	0	0	0	0	1	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	80%
انزل	0	0	0	0	0	0	0	0	0	0	0	0	**11**	1	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
اصعد	0	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	0	0	0	0	0	0	0	0	0	0	1	0	0	0	94%
اقرأ	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	80%
أكتب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	87%
تكلم	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	100%
اسكت	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
أجب	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	**12**	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
فوق	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	74%
ابدأ	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	**10**	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	67%
توقف	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	74%
أكمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	87%
امسح	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	87%
احمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
أنظر	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
انطلق	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0	100%
أعد	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	1	0	1	1	0	0	0	0	0	0	0	80%
نعم	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**10**	1	0	0	1	0	0	0	0	0	0	0	67%
لا	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	1	0	0	0	0	2	0	0	74%
صفر	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	1	0	0	87%
واحد	1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	74%
اثنين	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	0	0	100%
ثلاثة	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	0	0	1	0	0	74%
أربعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	3	0	0	80%
خمسة	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	94%
ستة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	100%
سبعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	100%
ثمانية	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	100%
تسعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	100%
TOTAL ACCURACY % = **87 %**																																									

In Table 5.5, the system based on MFCC features and Dynamic Time Warping method shows best performance with 90% recognition rate. Where the DTW will control the the time scale of the test speech signal to match the training time length taking into account the variation of the

speed when pronouncing a word. 13 words have full accuracy of 100% whereas the lower accuracy is 74% which is also acceptable.

Table 5-5 Confusion matrix of the system when using MFCC features and DTW classification

	أمام	خلف	يمين	يسار	أعلى	أسفل	تحرك	قف	أسرع	تمهل	افتح	أغلق	انزل	اصعد	اقرأ	أكتب	تكلم	اسكت	أجب	فوق	ابدأ	توقف	أكمل	امسح	احمل	أنظر	انطلق	أعد	نعم	لا	صفر	واحد	اثنين	ثلاثة	أربعة	خمسة	ستة	سبعة	ثمانية	تسعة	Accuracy%
أمام	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
خلف	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
يمين	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
يسار	0	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
أعلى	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	94%
أسفل	0	0	0	0	0	**14**	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
تحرك	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	87%
قف	0	0	0	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	94%
أسرع	0	0	0	0	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	94%
تمهل	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	80%
افتح	0	0	0	0	0	0	0	0	0	0	**13**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	87%
أغلق	0	0	0	0	1	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	80%
انزل	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
اصعد	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
اقرأ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	87%
أكتب	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	80%
تكلم	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	87%
اسكت	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	**12**	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
أجب	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**11**	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	74%
فوق	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	94%
ابدأ	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	**12**	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	80%
توقف	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
أكمل	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	80%
امسح	0	0	0	0	0	0	0	0	0	0	0	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	0	0	0	0	100%
احمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
أنظر	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	**12**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
انطلق	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	0	100%
أعد	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	100%
نعم	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**14**	0	0	0	0	0	0	0	0	0	0	0	94%
لا	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**11**	0	1	0	0	1	0	0	2	0	0	74%
صفر	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	1	0	0	0	80%
واحد	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	**12**	0	0	0	0	0	0	0	0	80%
اثنين	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	87%
ثلاثة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	**13**	0	0	0	0	0	0	0	87%
أربعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	0	100%
خمسة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	0	100%
ستة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	0	100%
سبعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	0	100%
ثمانية	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	0	100%
تسعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	0	100%
TOTAL ACCURACY % = **90%**																																									

In Table 5.6, the accuracy of the system based on LPC features with Itakura classification have a recognition rate of 83%. Where 2 words have the best accuracy of 100% and worst case is with the word "واحد" with accuracy of 60%.

Table 5-6 Confusion matrix of the system when using LPC features and Itakura classification

	أمام	خلف	يمين	يسار	أعلى	أسفل	تحرك	قف	أسرع	تمهل	افتح	أغلق	انزل	اصعد	اقرأ	أكتب	تكلم	اسكت	أجب	فوق	ابدأ	توقف	أكمل	امسح	احمل	أنظر	انطلق	أعد	نعم	لا	صفر	واحد	اثنين	ثلاثة	أربعة	خمسة	ستة	سبعة	ثمانية	تسعة	Accuracy%
أمام	**13**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	87%
خلف	1	**12**	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
يمين	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
يسار	0	0	0	**13**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	87%
أعلى	0	0	0	0	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	94%
أسفل	1	0	0	0	0	**12**	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	80%
تحرك	0	0	0	0	1	0	**13**	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
قف	0	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	80%
أسرع	0	0	0	0	0	1	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	80%
تمهل	1	0	0	0	0	1	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	74%
افتح	0	0	0	0	0	0	1	0	0	0	**12**	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
أغلق	0	0	0	0	0	0	1	1	0	0	0	**11**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	74%
انزل	0	0	0	0	0	0	0	1	0	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	80%
اصعد	0	0	0	0	0	0	1	0	1	0	0	0	0	**11**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	74%
اقرأ	0	0	0	0	0	0	1	0	0	0	0	0	0	1	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
أكتب	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	**11**	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
تكلم	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
اسكت	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	**13**	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
أجب	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	87%
فوق	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	**11**	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74%
ابدأ	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
توقف	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80%
أكمل	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	**11**	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	74%
امسح	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	87%
احمل	0	0	0	0	0	0	0	1	0	0	0	0	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	0	0	0	94%
أنظر	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	1	0	0	0	0	0	0	0	0	0	0	0	0	0	94%
انطلق	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	0	0	0	0	1	0	0	0	0	87%
أعد	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	0	0	0	0	0	0	0	80%
نعم	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	**12**	0	0	0	0	0	1	0	0	0	0	0	80%
لا	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**13**	0	0	0	0	1	0	0	0	0	0	87%
صفر	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	**11**	0	0	0	0	0	0	0	0	0	74%
واحد	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**9**	1	0	0	2	0	0	0	0	60%
اثنين	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**14**	0	0	0	0	0	1	0	94%
ثلاثة	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	**10**	0	0	0	0	0	0	67%
أربعة	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	0	0	80%
خمسة	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	**12**	0	0	0	0	80%
ستة	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	0	80%
سبعة	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**12**	0	0	80%
ثمانية	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	**13**	0	87%
تسعة	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**15**	100%
TOTAL ACCURACY % = **83%**																																									

We see from the previous tables that the correct rates of the system is 85.23%, 57% , 87%, 90%, 83% when using MFCC+Euclidean , Formants+DWT , MFCC+GMM , MFCC+DWT and LPC+ Itakura respectively. The worst correct rate is with Formants and the best one is with MFCC and using Dynamic Time Warping classification.

Also we see from the Figure 5.4, that MFCC+DWT (M4) outperforms the other methods for most of the words and only for few words MFCC+GMM (M3) outperform M4 and it is clear from the graph that the worst method is Formants+DWT (M2) for almost all the words.

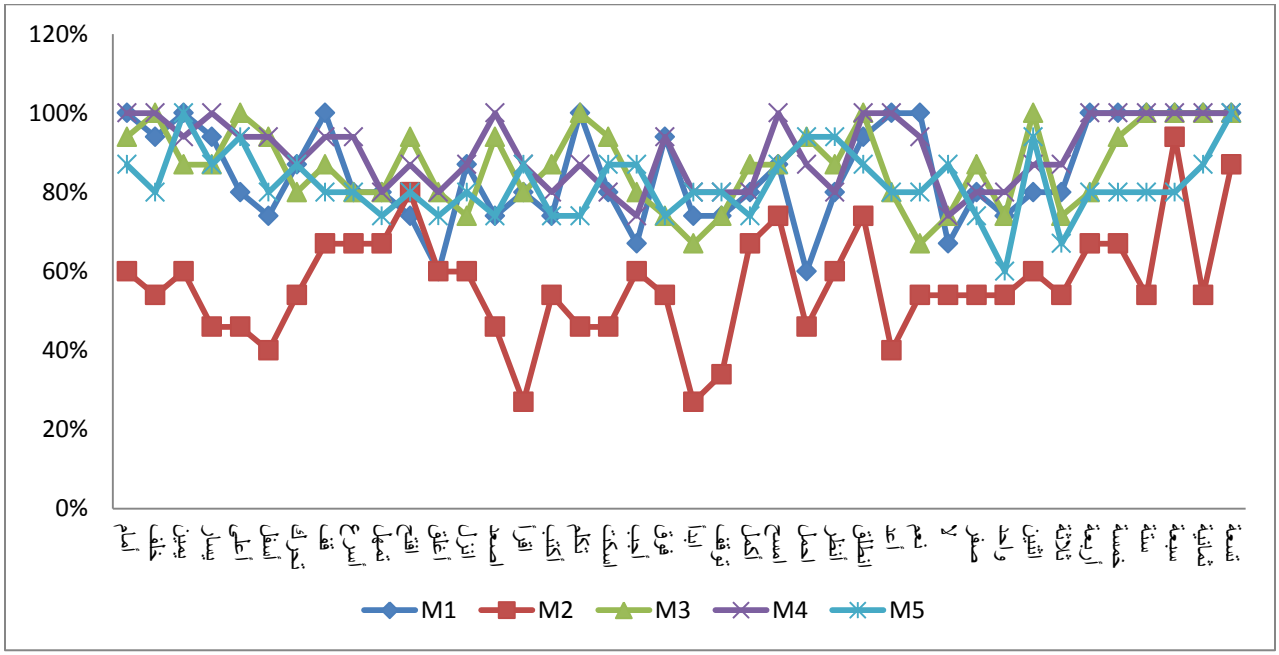

Figure 5.4 Words accuracy by the diffferent methods

Combination of the Methods

In the following table we will compare the perfomance (accuracy and execution time) of the different combination of the 5 methods using a voting rule. Where, the recognized item is the one that is recognized by the maximum number of methods. When no match between the methods, then we take the output of the best single method in the combination.

Let:

M1: Euclidean Classification with MFCC features (MFCC | Euclidean)

M2: DTW Classification with Formants features (Formant | DTW)

M3: GMM Classification with MFCC features (MFCC | GMM)

M4: DTW Classification with MFCC features (MFCC | DTW)

M5: Itakura Classification with LPC features (LPC | Itakura)

$M_i + M_j$: plus sign means combining the two methods

$(M_i + M_j) \rightarrow M_k$: means that the system will combine only M_i and M_j and when they did not give same classification then it will add M_k to the combined classifier.

Also, we need to combine at least 3 methods. Since combining 2 methods will have same output of the best single method. For example M1+M2:

If M1 output= M2 output then M1+M2 output= M1 output.

If M1 output ≠ M2 output then M1+M2 output= M1 output (will take the output of best single method).

Table 5-7 Perfomance of the different combinations of the 5 methods

Method Combinations	Average Accuracy	Average Computation Time(second)
M1	85.23%	0.7
M2	57%	0.3
M3	87%	0.2
M4	90 %	2.3
M5	83%	0.6
M1+M2+M3	85.73%	0.8
M1+M2+M4	92.33%	2.6
M1+M2+M5	86.94%	1.4
M1+M3+M4	92.5%	2.6
M1+M3+M5	90.27%	1.5
M1+M4+M5	93.83%	2.9
M2+M3+M4	92.39%	2.4
M2+M3+M5	89.94%	0.9
M2+M4+M5	92.39%	2.7
M3+M4+M5	93.72%	2.7
M1+M2+M3+M4	93.22%	2.6
M1+M2+M3+M5	90.72%	1.5
M1+M2+M4+M5	**94.39%**	2.9
M1+M3+M4+M5	93.60%	2.9
M2+M3+M4+M5	93.94%	2.7
M1+M2+M3+M4+M5	93.39%	3

From Table 5.7, we see that the best combination is M1+M2+M4+M5 (MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura) with an accuracy of 94.39% but its time computation is the largest 2.9 seconds. Also, MFCC with Gaussian mixture method is the fastest method with only 0.2 second but when it is combined with other methods does not give best result. This is due that our training data is not big enough and in our experiment, if we increase the training data this will increase the execution time too much, which is not suitable in our combination system case.

Also, we notice that such a combination can degrade the performance as in M1+M2+M3 the combined recognition rate is 85.73% is lower than the accuracy of M3 alone with 87%. For example if M3 has correct output whereas, M1and M2 have the same wrong outputs. Then the output of the voting system will be wrong even that M3 is correct.

Also we notice that the best single method is MFCC feature with Dynamic time warping but it is the most time consuming of all the single methods.

Since M1+M2+M4+M5 is the best method. We will select this combination and we will try to reduce the time computation by combining only two methods and when they do not match we will add another method to the combination. Also, we need to make the method M4 in the last decision of the combination since it is the most time consuming.

Table 5.8 shows the sub combination of M1+M2+M4+M5 to find the best accuracy and time computation.

Table 5-8 Subcombination of M1+M2+M4+M5 performances

	Average Accuracy	Average Computation Time(second)
M1+M2→M5+M4	93.56 %	1.55
M1+M5→M2+M4	**94.56 %**	**1.56**
M2+M5→M1+M4	92.9%	1.75
M1+M2+M5→M4	92.7 %	1.53

From the above table we find that the best one is M1+M5→M2+M4. Where first the system will combine the two fast methods M1 and M5 (MFCC | Euclidean + LPC | Itakura) and only when the two methods do not match the system will add other

combination M2+M4 (Formant | DTW + MFCC | DTW). We notice that the average time computation of the datasets is reduced to the half and is less than the time of the single method M4 alone. Since M1+M5 have a match in 26 words and consumes 0.8 second whereas only 14 words will use M1+ M2+ M4 + M5 which consumes 2.9 second.

The positive effect of combination method on the recognition rate is clearly observed in Figure 5.5, where best single method M4 has 90% and the combination of the methods improve the accuracy significantly to 94.56%. This is due that features combination adds an important speech parameters. Where MFCC gives some of the features of the words and the Formants and LPC give other features and combining them together will add more information of the words. Also, when using different classification method it improves the accuracy since the two methods will give the same classification to the word only when it has a high probability to be correct classification.

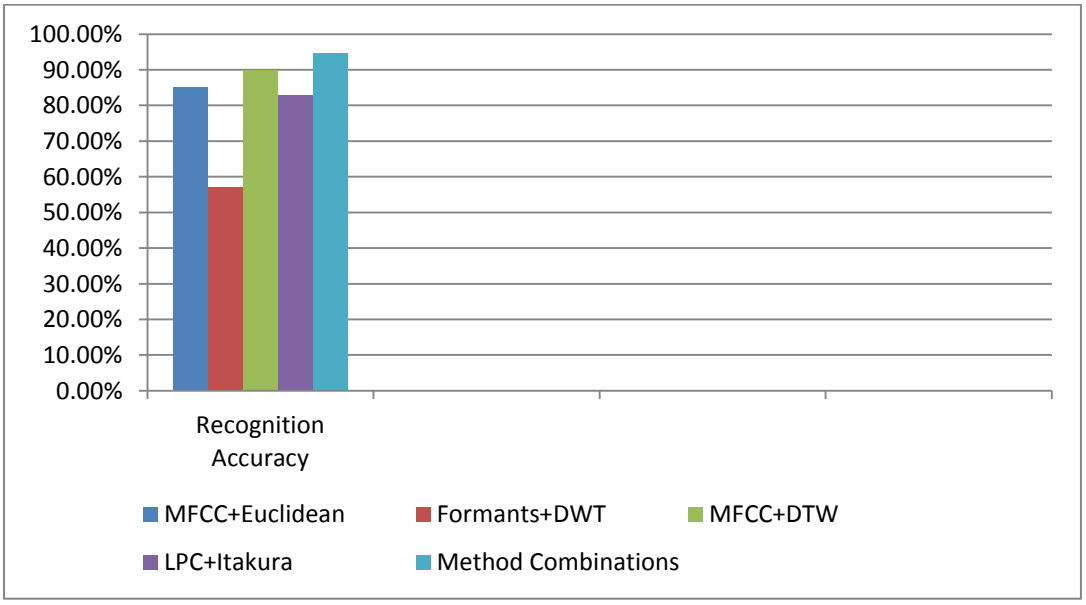

Figure 5.5 Recognition accuracy for different methods

5.4 Comparison With Other Researches

In this section we will try to compare our proposed system with similar systems in previous researches that use features or classifications combinations.

Bourouba et al. [11] presented a new Arabic digit recognition system based on classifier combination of HMM and a supervised classifier (SVM or KNN) with MFCC and the log

energy and pitch frequency feature extraction combination method. They found that using HMM classifier alone the accuracy is 88.26% and improved with the combined system to 92.72%. In [30], an Arabic digit recognition system was proposed based on MFCC+Δ + ΔΔ + log(energy). The system was developed using the Hidden Markov Models (HMMs) with vector quantization (VQ) and Tree distribution approximation model. The Dataset consists of 8800 samples (10 digits x 10 repetitions x 88 speakers). They get a good accuracy of 98.41%. In [31], an Arabic isolated word recognition system was proposed based on MFCC, log(energy) and their first and second derivatives. They used a Dynamic Time Warping (DTW) for the classification. The dataset consists of 19 words (digits from 1 to 10 plus 9 words) this word is uterred 3 times by 30 speakers a total of 1710 sample is the size of the dataset. They cited in their paper that the recognition accuracy was about 98.5% in a clean environment. In [32], an Arabic isolated word recognition system was proposed based on LPC and LPCC (LP cepstral coefficients)+ Delta LPC (the first derivative) using vector quantization (VQ) and HMM classification . The dataset used is composed of 1500 samples (50 speakers each of them uttered three times the ten digits). The recognition accuracy was about 91%. In [33], HMM-based Arabic isolated words recognition system was proposed using Wavelet cepstral coefficients and Mel frequency cepstral coefficients using a 500 samples datasets. The recognition accuracy was about 88.46%. In [34], a comparison of discrete Hidden Markov Model with vector quantization and DTW techniques was made for recognizing isolated words in Arabic language. The system is based on combined features MFCC, Energy and differential information (Δ and ΔΔ). The dataset consists of 500 samples (5speakers* 10 digits*10repititions). The better recognition accuracy of about 92% was obtained with DHMM-based system.

Table 5.9; summerizes the recognition rates obtained from the previous approaches. By comparing our system with the previous researches we conclude that our proposed system is very good and competitive to the other approaches. In our system we used 40 Arabic words whereas the others have used only 10 digits and only one with 19 words. Also, In order to have ideal comparison we need to have common database and same computer and software properties and with clear environment.

Table 5-9 Comparisons with previous researches

Paper Title	Features	Data Type	Classification Methods	Dataset	Recognition Accuracy
New Hybrid System (Supervised Classifier/Hmm) For Isolated Arabic Speech Recognition[11]	MFCC + log energy + pitch	Arabic digits	HMM + SVM /KNN	920 samples (10 digits x 92 speakers)	92.72%
The second-order derivatives of MFCC for improving spoken Arabic digits recognition using Tree Distributions pproximation Model and HMMs[30]	MFCC+ Log(energy) + (Δ and ΔΔ)	Arabic digits	HMMs+VQ	8800 samples (10 digits x 10 repetitions x 88 speakers)	98.41%
Efficient DTW-Based Speech Recognition System for Isolated Words of Arabic Language[31]	MFCC+ Log(energy) + (Δ and ΔΔ)	Arabic words and digits	DTW	1710 samples(30 speaker x 19 words x 3 repititions)	98.5%
Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition[32]	LPC + LPCC + Delta LPC	Arabic digits	VQ+HMM	1500 samples (50 speakers x 3 repitition x 10 digits).	91%
Multi-band based recognition of spoken Arabic numerals using wavelet transform[33]	Wavelet + MFCC	Arabic digits	HMM	data set consists of 500 utterances by 50 speakers	88.46%.
A Comparison of DHMM and DTW for Isolated Digits Recognition System of Arabic Language[34]	MFCC+Energy+ (Δ and ΔΔ)	Arabic digits	DTW+DHMM	500 samples (5 speakers x 10 digits x 10 repititions)	92%
Our Proposed System	MFCC+ LPC+ Formants	Arabic words and digits	Euclidean+ DTW+ Itakura	600 Samples (5 speakers x 40 digits x 3 repititions)	94.56 %

Chapter 6
Conclusion

6.1 Summary and Concluding Remarks

In this thesis, we designed a new speaker dependent isolated Arabic word speech recognition system based on a combination of several features extraction techniques MFCC, LPC, Formants and a hybrid of classification methods with Euclidean distance and Dynamic Time Warping. Where, the system combines the methods outputs using a voting rule. First, we used in the preprocessing step a word boundary detector using the energy and the zero crossing rates to automatically remove silences and identify the start and the end of the word in the input signal. Then discrete wavelet transform to the speech signal is performed before extracting the features to improve the accuracy of the recognition and to make the system more robust to noise.

In this thesis we compare 5 different methods which are MFCC+Euclidean, Formants+DTW, MFCC+GMM, MFCC+DTW and LPC+Itakura and we get a recognition rate of 85.23%, 57%, 87%, 90%, 83% respectively. Where MFCC+DTW has the best performance with 90% recognition rate which is an expected result since the MFCC is one of the best feature extraction techniques based on human hearing system and the Dynamic Time Warping is used to align words signals when measuring similarity to cope with different speaking speeds but this method has the worst execution time of 2.3 seconds. Whereas, the Formants+DTW method has the lowest accuracy of 57% but good exection time of 0.3 seconds.

In order to improve the accuracy of the system, we tested several combinations of these 5 methods. We find that the best combination is MFCC | Euclidean + Formant | DTW + MFCC | DTW + LPC | Itakura with an accuracy of 94.39% but its time computation is the largest 2.9 seconds. Where MFCC+GMM do not improve the combined system which is due to the limitation of GMM that it requires a sufficient amount of training data which may increases the execution time. Also, we find that some combination can degrade the performance of the system. Also, we conclude that Formants based method

should not be used alone since it has bad recognition rate but when combined with MFCC and LPC features the accuracy is improved significantly and outperforms MFCC based methods.

In order to reduce the computation time of this hybrid, we compare several subcombination of this hybrid and we find that the best performance in trade off computation time is with the system combining MFCC | Euclidean + LPC | Itakura and only when the two methods do not match the system will add the other combination Formant | DTW + MFCC | DTW. Where the average computation time is reduced to the half is 1.56 seconds and the system accuracy is improved become 94.56%.

In order to accelerate the system and to reduce the execution time, we first make the system to identify the speaker and load only the reference model of that user.

Comparing our proposed system with previous research shows that our system is very good and competitive to the other approaches. Also, our project is a multi-user system can be implemented in a single device and used by different persons. Since different user can store their reference template and the system can recognize the speaker and load its reference model for recognition.

The system is implemented with a graphic user interface under Matlab using G62 Core I3/2.26Ghz processor laptop. The dataset used in this system include 40 Arabic words recorded in a calm environment with 5 different speakers using laptop microphone. Each speaker will read each word 8 times. 5 of them are used in training and the remaining are used in the test phase. The datasets of the system are recorded in a calm environment and the words are spoken with same style and same distance to microphone to have same loudness.

Also, we find that some words have low accuracy this due to the noise in some test data or the variation of speaking style and rate between the training and the testing data and due to the large similarity between the pronunciations of certain words. The performance could be improved by training the system with large datasets.

.

Finally, the system accuracy is improved becomes 94.56% when using combination of the methods. Hybrid methods try to reduce their limitations by combining the advantages of the combined techniques. Where MFCC give some of the features of the words and the Formants and LPC give other features and combining them together will add more information of the words. Also when using different classification method it improves the accuracy since the two methods will give the same classification to the word only when it has a high probability to be correct classification. Hybrid system is one of the emerging approaches that can improve speech recognition accuracy and will take an important role in future speech technology research.

6.2 Future Work and Recommendations

In the future work we will try to improve our system to move to the general case of independent speaker large vocabulary Arabic continuous speech recognition system that is robust to noise and to the differences in speech style and loudness. Also we will try the following techniques:

- Developing the system by using other noise cancellation techniques.
- Improve the performance of the system by using large training datasets.
- Extend the work to include more words.
- Select other features extraction techniques such as the image of the spectrum of the speech and we will use image techniques to extract the features of the speech spectrum.
- Combine other different methods to improve accuracy in trade off computation time.

Since the system is sensitive to noise and the accuracy will be decreased in the presence of noise. We recommand to buy a noise-cancelling microphone that filter the background noise and that has logarithmic sensitivity to distance to not include far distances noise to the processing or we may use multi microphone system to better isolate speech from noise. Also, we recommand to build a common arabic database, in order that researcher benefit from it and do not spend time in recording and gathering database and also to be able to compare their approach with same database.

References

[1] E. Mengusoglu, "*Confidence Measures for Speech/Speaker Recognition and Applications on Turkish LVCSR*", *PhD. thesis*, 2004.

[2] K. Kirchho , J. Bilmes, J. Henderson, R. Schwartz, M. Noamany, P. Schone, G. Ji, S. Das, M. Egan, F. He, D.Vergyri, D. Liu and N. Duta, "*Novel Approaches to Arabic Speech Recognition*", Technical Report, Ohns-Hopkins University, 2002.

[3] D. Vergyri and K. Kirchhoff, "*Automatic Diacritization of Arabic for Acoustic Modelling in Speech Recognition*", Editors, Coling, Geneva, 2004.

[4] D. Vergyri, K. Kirchhoff, K. Duh and A. Stolcke, "*Morphology Based Language Modeling for Arabic Speech Recognition*", in Proceedings of Interspeech, Germany, pp. 2245-2248, 2004.

[5] H. Choiy, R. Gutierrez, S. Choiz and Y. Choe ,"*Kernel Oriented Discriminant Analysis for Speaker-Independent Phoneme Spaces*", icpr 2008.

[6] A. Mishra, M. Chandra, A. Biswas, and S. Sharan, "*Robust features for connected Hindi digits recognition*", Intl. Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 4, no. 2, pp. 79-90, June 2011.

[7] M. Nilsson and M. Ejnarsson. "*Speech Recognition using Hidden Markov Model: Performance evaluation in Noisy Environment*", Master thesis, Department of telecommunications and signal processing, Blekinge Institute of technology, Sweden, March 2002.

[8] B. Al-Qatab and R.Ainon, "Arabic Speech Recognition Using Hidden Markov Model Toolkit(HTK)", IEEE, International Symposium on Information Technology, ITSim, vol. 2, pp. 557-562, 2010.

[9] M. Elshafei, H. Al-Muhtaseb and M. Al-Ghamdi, "*Speaker-independent Natural Arabic Speech Recognition System*", The International Conference on Intelligent Systems ICIS 2008, Bahrain, December 2008.

[10] M. El Choubassi, H. El Khoury, C. Alagha, J. Skaf and M. AlAlaoui, "*Arabic Speech Recognition Using Recurrent Neural Networks*", IEEE Intl. Symp. Signal Processing and Information Technology ISSPIT, 3(1), 336-340, 2003.

[11] H. Bourouba, R. Djemili *et al*, "*New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition*", 2nd Information and Communication Technologies, ICTTA-06, 2006.

[12] M. Shoaib, F. Rasheed, J. Akhtar, M. Awais, S. Masud and S. Shamail, "*A novel approach to increase the robustness of speaker independent Arabic speech recognition*", 7th international multi topic conference, INMIC, pp 371–376, 2003.

[13] A. Amrous, M. Debyeche and A. Amrouche, "*Integration of Auxiliary Features in Hidden Markov Models for Arabic Speech Recognition*", International Conference on Signals, Circuits and Systems, 2009.

[14] J. G. Proakis and D. G. Manolakis, "*Digital Signal Processing. Principles, Algorithms and Applications*", 3rd Edition, Macmillan, New York, 1996.

[15] E. Essa, A. Tolba and S. Elmougy, "*A Comparison of Combined Classifier Architectures for Arabic Speech Recognition*", in the Proceedings of the 2008 IEEE International Conference on Computer Engineering & Systems, Cairo, November 2008.

[16] Akram M. Othman, and May H. Riadh, "*Speech Recognition Using Scaly Neural Networks*", World Academy of Science, Engineering and Technology 38, 2008.

[17] Y. Alotaibi, M. Alghamdi and F. Alotaiby, "*Speech Recognition System of Arabic Digits based on A Telephony Arabic Corpus*", International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV'08), Las Vegas, 2008.

[18] E. Hagos, "I*mplementation of an Isolated Word Recognition System*"; UMI Dissertation Service, 1985.

[19] W. Abdulah and M. Abdul-Karim, "*Real-time Spoken Arabic Recognizer*"; Int. J. Electronics, Vol. No. 59, Issue No. 5, pp.645-648, 1984.

[20] A. Zolnay, R. Schlüter, and H. Ney, "*Acoustic Feature Combination for Robust Speech Recognition*", In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), volume 1, pages 457-460, Philadelphia, PA, March 2005.

[21] E. Essa, A. Tolba and S. Elmougy, "*Combined Classifier Based Arabic Speech Recognition*", in the Proceedings of the 6th International Conference on Informatics and Systems (INFOS2008), Cairo, March 2008.

[22] M. A. Anusuya, S. K. Katti, "*Speech Recognition by Machine: A Review*" International journal of computer science and Information Security, 2009.

[23] M. A. Anusuya and S. K. Katti, "*Classification Techniques used in Speech Recognition Applications: A Review*", Int. J. Comp.Tech. Appl., Vol 2, 4, pp 910-954, 2011.

[24] B. S. Atal and M. R. Schroeder, "*Predictive coding of speech signals*", in Report of the 5th Int. Congress on Acoustics, 1968.

[25] A. M. Toh, "*Feature Extraction for Robust Speech Recognition in Hostile Environments*", *PhD. thesis, Dept. Elect. Eng., Western Australia Univ.,Australia , 2007.*

[26] J. Park, F. Diehl, M. Gales and M. Tomalin, "*Training and adapting MLP features for Arabic speech recognition*", In: Proc. of ICASSP, pp. 4461–4464, 2009.

[27] Z. Y. Mohammed and A. M. Khidhir, "*Real-Time Arabic Speech Recognition*", International Journal of Computer Applications (IJCA), Volume 81, No.4,pp.:43-45, Nov. 2013.

[28] N.Hammami, M.Bedda and N. Farah, "*Spoken Arabic Digits recognition Using MFCC based on GMM*", IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, pp.:160 – 163, Oct. 2012.

[29] K. Fujinaga, M. Nakai, H. Shimodaira, and S. Sagayama, "*Multiple-regression hidden Markov model*", In Proc. IEEE Intl. Conf. on Acoust. Speech & Signal Process., ICASSP, pages 513–516, Salt Lake City, USA, May 2001.

[30] N.Hammami, M.Bedda and N. Farah, "*The second-order derivatives of MFCC for improving spoken Arabic digits recognition using Tree Distributions Approximation Model and HMMs*", In Proc. IEEE Intl. Conf. on Communications and Information Technology (ICCIT), pp. 1-5, 2012.

[31] K. A. Darabkh, A. F. Khalifeh, I. Jafar, B. A. Bathech and S. W. Sabah, "*Efficient DTW-Based Speech Recognition System for Isolated Words of Arabic Language*", in the proceedings of the International Conference on Electrical and Computer Systems Engineering (ICECSE2013), Switzerland, May 2013.

[32] H. Bahi and M. Sellami, "*Combination of Vector Quantization and Hidden Markov Models for Arabic Speech Recognition*", Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Beirut, Lebanon, pp: 96-100, June 2001.

[33] W. Alkhaldi, W. Fakhr and N. Hamdy, "*Multi-Band Based Recognition of Spoken Arabic Numerals Using Wavelet Transform*", Proceedings of the 19th National Radio Science Conference (NRSC'01), Alexandria University, Alexandria, Egypt, March 19-21, 2002.

[34] Z. Hachkar, A. Farchi, B. Mounir and J. El Abbadi, "*A Comparison of DHMM and DTW for Isolated Digits Recognition System of Arabic Language*", International Journal on Computer Science and Engineering, vol.3, no.3, pp.1002-1008, March 2011.

[35] M. D. Al-Hassani and A. A. Kadhim, "*Design A Text-Prompt Speaker Recognition System using Lpc-Derived Features*", International Journal of Information Technology and Management Information Systems (IJITMIS), Volume 4, Issue 3, pp. 68 - 84, ISSN Print: 0976 – 6405, ISSN Online: 0976 – 6413, 2013.

[36] D. Sripathi, "*Efficient Implementations of Discrete Wavelet Transforms Using FPGAs*", M.S. thesis, Florida State University, 2003.

[37] L. Deng and J. Dang, "*Speech analysis: the production-perception perspective*", in In: Lee, Chin-Hui, et al. Advances in Chinese spoken language processing. Singapore; Hackensack, N.J, pp. 3-32, World Scientific, 2007.

[38] A. S. Parihar, "*Fuzzy Filter Design For Noise Removal In Color Image Using Wavelet*," M.S. thesis, Dept. Elect. Eng., Delhi Univ., India, 2008.

[39] M. D. Al-Hassani, "*Identification Techniques using Speech Signals and Fingerprints*", Ph.D. Thesis, Department of Computer Science, Al-Nahrain University, Baghdad, Iraq, September 2006.

[40] K. Li, M. Fei, G. W. Irwin, S. Ma, "*Bio-Inspired Computational Intelligence and Applications*", International Conference on Life System Modeling and Simulation, LSMS, Shanghai, China, 2007.

[41] A. Głowacz and W. Głowacz, "*New Approach to Diagnostics of DC Machines by Sound Recognition Using Linear Predictive Coding*", Advances in Intelligent and Soft Computing Volume 60, pp 529-540, 2009.

[42] B. HATİPOĞLU, "*A Wireless Entryphone System Implementation with MSP430 and CC1100*", Dept. Comput. Eng., Yeditepe Univ., Rep., 2008.

[43] A. J. Jerri, "*The Shannon sampling theorem- its various extensions and applications: a tutorial review*", Proc. IEEE, 65, no. 11 (1977) 1565-1596.

[44] C. Yılmaz, "*A Large Vocabulary Speech Recognition System for Turkish*", MS Thesis, Bilkent University, Institute of Engineering and Science, Ankara, Turkey, 1999.

[45] I. Feigler, "*Time frequency analysis of ECG signals*", Dept. Science, Masaryk Univ., Rep., 2013.

[46] B. Pellom, "*Automatic Speech Recognition : From Theory to Practice*", Dept. Comput. Science, Colorad Univ., Rep., 2004.

[47] C. Becchetti and K. Ricotti, "*Speech Recognition*", Choudhary Press Delhi, ISBN 978-81-265-1774-9, 2008.

[48] T. Kinnunen, T. Kilpeläinen, P. Fränti, "*Comparison of clustering algorithms in speaker identification*", Proc IASTED Int Conf Signal Process Commun, 2000.