PALESTINE POLYTECHNIC UNIVERSITY
COLLEGE OF INFORMATION TECHNOLOGY AND COMPUTER ENGINEERING

Embedded Quran Voice Recognition and Tracking System

Team members:
Ahmed Hashlamon
Mohammad Altakrouri

Supervisor:
Dr. Radwan Tahboub

Co-Supervisor:
Prof. Dr. Karim Tahboub

May 20th, 2015

Abstract

The art of Tajweed and linguistic properties of Arabic has always been an interesting field for research. However, it's a new research field with computer science. This project represents the problem of Quran recitation verification for Quran readers and Imams around the world. In this project real time Quran speech recognition will be used to solve this problem using the "Banana Pi" microcontroller. The process will start by collecting Quran speech from the reader using a microphone. After sampling the signal, preprocessing should be done to the signal, and then features vectors that will be extracted to be matched with the most similar phrase from the database, where the recognition will be using efficient and reliable algorithm which is "Dynamic Time Wrapping" - DTW. The result will be displayed on a monitor connected to the microcontroller with a notification if needed in the case of making mistakes.

إهداء

إلى أمهاتنا اللاتي ركع العطاء أمام أقدامهنّ وأعطونا من دمائهنّ، أرواحهنّ وأعمارهنّ حباً وتصميماً ودافعاً لغدٍ أجمل.. إليكنّ يا من لا نرى الأمل إلا في أعينكنّ...

إلى آبائنا الذين زرعونا أشجاراً خضراءَ وأسقونا الرّضا، وها هم الآن يقطفون ثماراً كانوا هم غارسوها.. إليكم يا من لا تفيكم كلمات الشكر والعرفان بالجميل...

إلى الأرواح الخفية التي لطالما وقفت الى جوارنا وأخذت بأيدينا إلى خطوة تالية أجمل.. إليكم شكرنا الجزيل وامتناننا...

إلى هذه الأرض الطّهور ثراها، وإلى كل شُهدائنا الأبرار الذين بذلوا أرواحهم فداء لنا وللقُدس، إلى كل أسرانا الذين يقبعون في غياهب السّجون ظُلماً وقهراً.. إلى قُدسِنا الحبيبة..

إلى كُلّ الذينَ عبروا في حياتنا وتركوا الأثر، وكلّ اللذينَ لم يعبروا لحكمةٍ أعدّها القدر.. نجاحاتنا تُتوّج بكُم...

Acknowledgments

All praise is due to Allah, the Creator and Sustainer of this whole universe, the Most Beneficent and the Most Merciful, for his guidance and blessing and granting us knowledge, patience and perseverance to accomplish this project successfully.

Firstly, we would like to acknowledge Palestine Polytechnic University represented by our instructors, especially College of Information Technology and Computer Engineering for providing us support to carry out this project. We take great pride to forward our sincere appreciation and deepest gratitude to our supervisor Dr. Radwan Tahboub, and Co-supervisor Prof. Dr. Karim Tahboub for their valuable guidance, support, encouragement and effort throughout this graduation project all these months. Without their tireless efforts, patience and guidance, this project could not have been successfully completed.

Secondly, we are also grateful to Eng. Sa'adi Al-Tamimi for provision of expertise, and technical support in the implementation. And great thanks to all special people who helped and supported us all from their positions. Without them, the project would not be like in this quality of outcomes, and thus their support has been essential.

Last but not least, most profound gratitude and respect to our families, especially our beloved parents, who have been the ultimate source of our motivation to work hard and inspiration of our life. Therefore, we proudly dedicate this work to all of them, may Allah bless them.

May 2015

Ahmed Hashlamon, Mohammad Altakrouri,
Department of Computer Engineering,
College of Information Technology and Computer Engineering,
Palestine Polytechnic University,
Palestine.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

This chapter presents an overview of the project, including the project idea description, the problem statement, project objectives, system requirements, scope of the project, motivations, methodology, and project documentation outline.

1.1 Overview

By the year 2014 as the world became so connected to technology in everything, people have recently preferred the automated systems in their Smartphones, tablets, laptops, or any technological tool that would ease accomplishing their needs and save time.

As this project has been produced from the decision to do something related to our religion, culture and Muslims all around the world, a raised problem –which is faced by many Muslim prayers, especially the Imam while reciting in aloud prayers- is that it's possible to forget the next words in a verse, or the next verse, or one might jump to another similar verse in another sourate or page. Moreover, there is the need for a modern and technological approach for self-learning methods to recite the Quran, which can improve the learning process of the Quran and also optimize the study time of the students.

Since speech is considered as the most widely used and natural means of communication between humans, it's an obvious substitute for such means of keyboard or interactive screens in the communication process. Regarding this project, many IT tools may provide technological help and solve such a problem by receiving the reader's voice and processing it.

So, the aim of our project is to develop an automatic speech recognition system that recognizes sites of verses the reader is reading, with input of Imam Quran recitation and output of the read Quran page displayed on a screen in front of him.

The discussion will mainly be about speech recognition –a part of signal processing technology, which is highly demanded for many useful applications-, in addition with various components from artificial intelligence that incorporates with the recognition needs.

1.2 Project idea description

The project idea is to build and develop a Quran voice recognition system that listens to the Quran reader's voice while reading, detect in which page and verse the reader is reading, and display that page on a display screen. If the reader makes any mistake while reading, the system should detect the mistake and correct the reader if he skips to another verse or page. However, the system is implemented on Banana Pi microcontroller as an embedded system.

1.3 Project objectives

1. To help the Imam while reciting in aloud prayers from making mistake such as skipping to another or similar verse in Quran by developing a system that can recognize where he's reading and keep following with him.
2. To help Quran students to recite the Quran by an automated self-learning method, that will optimize the studying time and improve the learning process.
3. To develop an embedded system that combines feature extraction and recognition of Quran.
4. To define the best algorithm for Quran voice recognition.

1.4 System requirements

1. Collect the input speech, pre-process it and extract the features from it to produce a set of features vector.
2. To decide the best training tool to train the machine using the features vector against the stored features.
3. Match the unknown features vector produced from the input against the database to find the best match.
4. Display the best matched verse and the few following verses from the database.

1.5 Motivations

1. In the aloud prayers, the Imam is usually reading Quran by heart, and many Imam's make mistakes while reading, such as, jumping to another alike verse and page, or they may forget the next verse or some words, so our system will solve this problem by detecting where the Imam is reading and guide him.
2. The need for modern approaches to be used as self-learning method to recite Quran that would improve the learning process and optimize the study time.

1.6 Project scope:

This project is a research and system based project since its purpose is to build an embedded system using a microcontroller, and trying to make something new with the Quran speech recognition area by finding the best algorithm to give the satisfying match between input verses and the stored ones.

Signal processing and voice recognition are the main fields in this project, where the system will receive the input Quran voice from the reader through a microphone, and extract the feature form it to find the read verse and page through probability matching.

1.7 Problem statement

To build a small and cheap independent embedded system that can listen to the Imam in aloud prayers and detect the verse that he's reading, then display it on a screen after extracting the features and give the exact match from Quran verses features stored in the system's database.

1.8 Methodology

The work will depend on two major steps. Firstly -as a prerequisites- to have a good background in Signals and Systems theory, and to be related as much as possible with related researches and experiments that have been already published. Also it's required to have the knowledge in the Quran manner and tones, since the system will be implemented for such a specific and finite set of verses. Secondly; along with the system implementation on the hardware, MATLAB simulations will be hold to ensure the correctness of each step during the process.

1.9 Project outline

This project documentation contains 6 chapters, and below are the titles and summaries of each one:

Chapter 2 highlights the key of related researches, algorithms and techniques that are related and relevant with this research, in terms of commonly used of features extraction, classification and pattern matching techniques used, and provides an overview on the Raspberry Pi microcontroller and its features.

Chapter 3 provides an architecture design for Quran speech recognition system. The sub-topics for this chapter include the research design of this project and its implementation, as well as many other diagrams that represent the logical and physical designs of the systems.

Chapter 4 contains a description of the software used, flowcharts, algorithms used, and how the system runs.

Chapter 5 contains experimental data and results as well as other extra information, analysis and discussion of the result obtained after the training and testing procedures executed on the system. Also, an evaluation of the performance of the overall system.

Chapter 6 summarizes the work accomplished and discussed the possibilities and the recommendations for future work.

Chapter 2: Background

This chapter highlights the key of related researches such as the Arabic language, the art of Tajweed, and the linguistic properties of Arabic. It also highlights algorithms and techniques that are related and relevant with this research, in terms of commonly used features of extraction, classification and pattern matching of techniques used. Also to provide an overview on the Banana Pi microcontroller and its features.

2.1 Arabic language:

Recently, speech recognition has become one of the top topics in signal processing, and many useful applications rely on it. The main area of this project is Quran speech recognition, which is in the Arabic language [1].

Arabic is a language that is widely used, it's a mother tongue for more than 350 million people in the world. Also, hundreds of millions speak passive Arabic to read the Quran. However, it's considered as one of most complex languages in the world [2, 3].

As we know, each person has a different voice, thus, the input speech could differ between any speaker and another according to many factors. Here in Arabic we have some important factors which are unique such as the Harakat and prolongation which could make any sentence different than from itself according to how the speaker pronounced it and produces another sound.

2.2 The art of Tajweed:

"Tajweed" is an Arabic word meaning proper pronunciation during recitation, as well as recitation at a moderate speed. It is a set of rules which govern how the Quran should be read [1]. The Tajweed follows many flexible rules and different ways of reading called "Qiara'a"; they're 10 basic recitation ways for the Quran like: Hafs, Kaloun, Warsh…, all are correct [2].

Moreover, Tajweed has a set of rules called "the Ahkam" such as Idgham, Ithhar, Ikhfa', and Ghunna. Most of these rules should be applied by the reader while reading. Thus, all of these combinations of rules and ways of reading create a big difference between Arabic reading and Quran reading [4].

Also, in the art of Tajweed, there are different "Maqamat" in reading, used by the reciters to vary the tone of their recitation [4]. This also plays a role in varying the reader's voice which is another challenge for Quran voice recognition.

Here are the most important Tajweed rules affect the Quran recognition aspect [4]:

i. Necessary prolongation of 6 vowels.
ii. Obligatory prolongation of 4 or 5 vowels.

iii. Permissible prolongation of 2, 4 or 6 vowels.
iv. Normal prolongation of 2 vowels.
v. Nasalization (ghunnah) of 2 vowels.
vi. Silent unannounced letters.
vii. Emphatic pronunciation of the letter R.

Some echoing may be noticed in Quran recognition as a result of applying these rules, and can be prevented and canceled by filtering.

2.3 Linguistic properties of Arabic

The Arabic language has many features which make it a special and complex one as we mentioned earlier. It consists of 28 letters: 25 constants and 3 vowels (a, u, i) [5, 6].

Arabic letters are mostly connected, and there's no capitalization in Arabic such as English or other languages. Each letter of these letters may have 2 to 4 shapes: isolated shape, at the beginning of a word, in the middle, or at the end of a word [7].

Each one of these letters could be directed by a diacritic, or more called "Harakat" for setting up the grammatical functions and vary the pronunciation. There are 6 diacritics in Arabic which are:

i. Fatha: " َ " / a /
ii. Damma: " ُ " / u /
iii. Kasra: " ِ " / i /
iv. Sukon: " ْ " /silence/
v. Shadda: " ّ " / constant doubling /
vi. Tanween: the /n/ with Fatha, Damma, or Kasra

So, in this project the targeted space of Arabic language is for Quran recognition, moreover; some constraints should be considered such as choosing "Qira'atHafs" and reading with acceptable Tajweed skills.

2.4 Voice recognition:

Speech recognition is the translation of spoken words into text, which can be used for either identification or verification purposes. The process in figure (2.1) is described with brief details next.

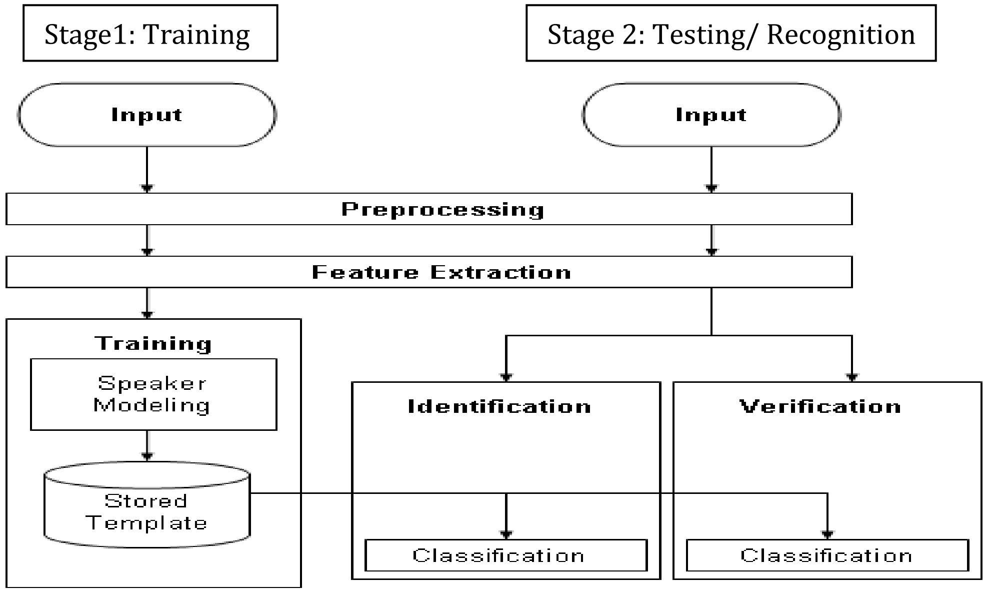

Figure2.1: Recognition system architecture

For such a recognition system, the following steps are required to do the process:

1- Pre-processing: used to simplify and improve the feature extraction, with a main procedure that can help as:
   a. Noise and Pre-Emphasis Filtering and Smoothing.

2- Feature extraction: the process of extracting measurements from the input to differentiate among classes [8], and here we're looking for unique, discriminative and robust signature to differentiate between each verse. Some of the techniques to extract features are:
   a. Linear Predictive Coding (LPC) coefficients extracted from speech tokens, converted to Cepstral coefficients and used along with neural networks. It is sensitive for noise, and still can be modified to reduce its sensitivity [9].
   b. Spectrographic Analysis: [10] implemented a strategy to extract a spectrogram for each Arabic phoneme depending on a specific frequency band. However, its problems are in the execution time and the difficulty to get automated.
   c. Mel-Frequency Cepstral Coefficient (MFCC): convert the speech waveform into the form of parametric representation, and it's one of the most popular techniques for feature extraction [11]. It's based on the frequency domain of Mel scale for human ear scale –according to human frequency bandwidth- [12]. When dealing with frequency domain, the system will be less sensitive for physical condition of speakers' vocal cord or the speech waveforms themselves [13]. The extraction process for MFCC is done by seven steps: Preprocessing, Framing, Windowing, DFT,

Mel-Filterbank, Logarithm, and Inverse DFT.

3- Training/ classification

Spoken language recognition is placed under the class of pattern recognition, and the next classification tools can be used for such a model:

a. Hidden Markov Model (HMM):

Implemented to recognize handwriting [14], and also used for speech recognition. The structure consists of

- States: the hidden part of the input.
- Transitions: links between the states.
- Observations: the known part of the input and it depends on states.

b. Artificial Neural Network (ANN)

A computational model based on the brain nervous system, simulating the nervous interconnecting to solve artificial intelligence problems. It consists of layers of nodes (neural) connected to each other, and the NN function defined as: a "class of functions" obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity [15].

c. Dynamic Time Warping (DTW)

An algorithm used to find the best alignment between two signals by applying a piecewise linear mapping of the time axis to align both signals, and in other words; it's a technique used to measure the similarity of two signals that vary in time or speed [16].

4- Recognition

Referring to the previous techniques, three ways are used to compare the output of the classification process with the saved data:

a. HMM: when comparing features, we take the highest matching probability according to a Gaussian distribution of the data set.
b. ANN: Vitrebi search [17] is applied to find the best path score.
c. DTW: using the Euclidean distance mapping function, the path that minimizes the distance between two words will represent the best match.

2.5 Banana Pi microcontroller

In this project, the whole processing of the voice recognition and feature matching will be on a microcontroller gets the input Quran voice from the microphone and displays it on a monitor.

Banana Pi is a single board powerful microcontroller that was developed early months of 2014; it can run different operating systems such as: Raspbian, Ubuntu, Android, and Debian.

As a microcontroller, it has the biggest RAM among other microcontrollers; its RAM is 1 GB, which is double of the new Raspberry Pi model B+. Also, it has a Dual Core 1 GHz, an SD slot up to 64 GBs, and2x USB 2.0 host, 1x USB OTG. It has a built in microphone, and it's able to output up to 1.6A, which means users can drive an external HDD without an extra power supply [18].

The monitor that will be used should have an HDMI interfacing to be compatible with the Banana Pi.

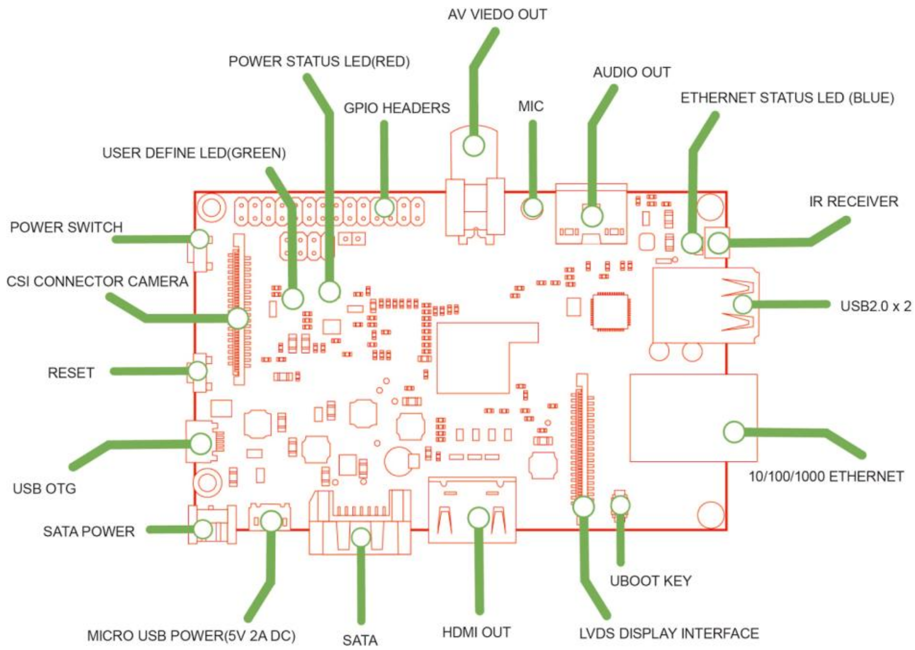

Figure2.2: Banana Pi Input/ Output (Front side)

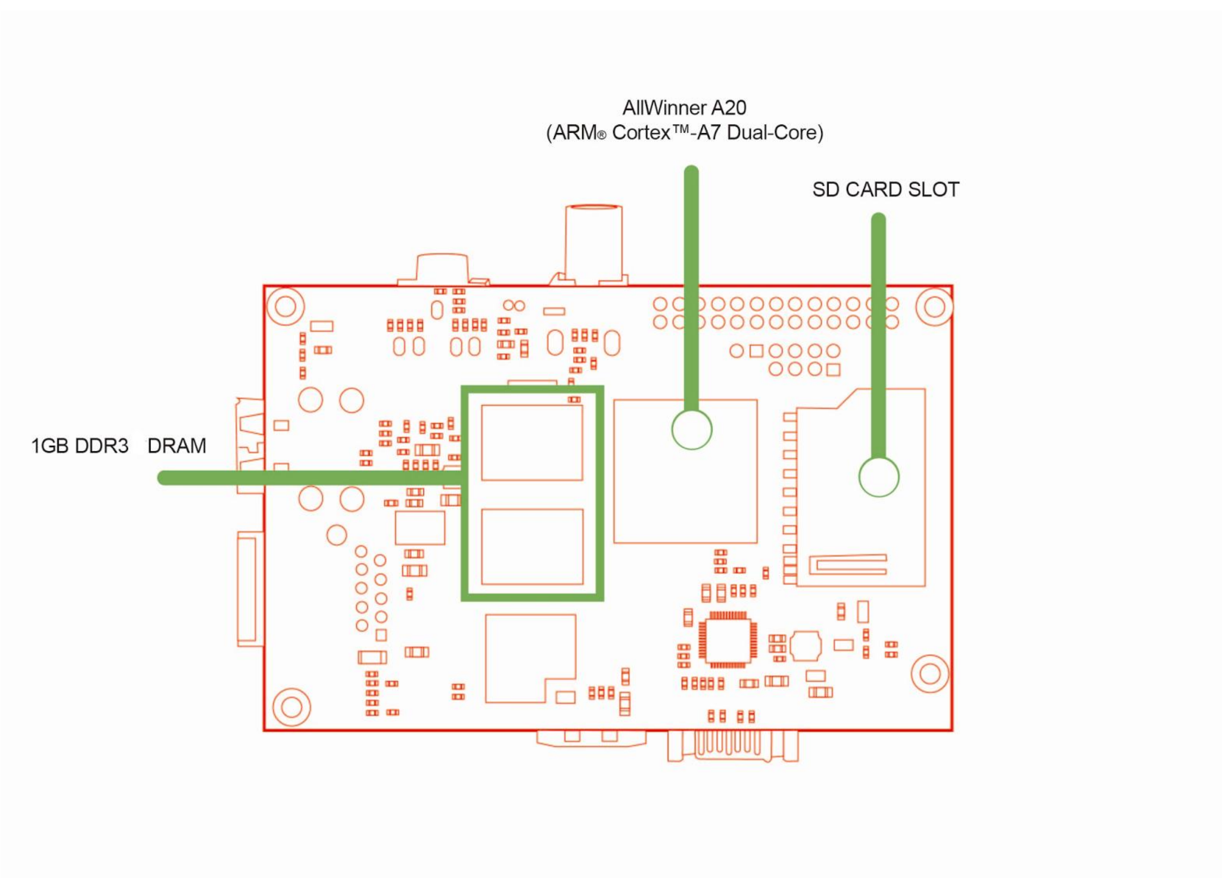

Figure2.3: Banana Pi Input/ Output (Back side)

Chapter 3: Analysis and Design

This chapter provides an analysis of the system, its detailed requirements, and an architecture design for Quran speech recognition system. The sub-topics for this chapter include the research design of this project and its implementation as pre-processing, feature extraction, and feature matching, as well as many other diagrams that represent the logical and physical designs of the systems.

3.1 System analysis

In this section and the following sections of this chapter the whole processing that happens inside the microcontroller, algorithms and techniques will be covered clearly.

First, the collected input Quran voice will be filtered and will get rid of the noise and echoing. After that, features will be extracted using Mel-Frequency Cepstral Coefficients (MFCC) feature extraction technique which will output a set of features vector represent the important characteristics and features in the Quran speech. Then, the read verse could be detected and classification could happen by one of the classifications options; Hidden Markov Model (HMM), or by Dynamic Time Wrapping technique (DTW).

3.1.1 Quran voice recognition system as physical components

Mainly, our system will consist of 3 components; a microphone collecting the Quran speech voice from the reader passing it to the Banana Pi microcontroller that will do the whole processing of the system and finally a monitor connected via HDMI cable to display the page that the reader reads figure (3.1).

Figure3.1: Overview of the Quran embedded system

3.1.2System detailed requirements

1- Collect the input speech via a microphone.
2- Filter the input speech from noise.
3- Digitize the filtered speech.
4- Extract the features from it to produce a set of features vector by MFCC.
5- Store the features vector in a database for DTW feature matching / build the HMM for each speech frame and store it in the database.

6- Feature matching: match the unknown features vector produced from the input against the features vector stored in the database to find the matches, or against the HMM models stored.
7- Decide the best match as the highest probability match of features vectors in the database.
8- Display the best match verse and the few following verses from the database.

3.2System design

In order to develop the Quran voice recognition system and generally in voice recognition, the following techniques specified have to be followed:

1. Pre-Processing.
2. Feature Extraction.
3. Feature Classification / Training.
4. Recognition/ Identification.

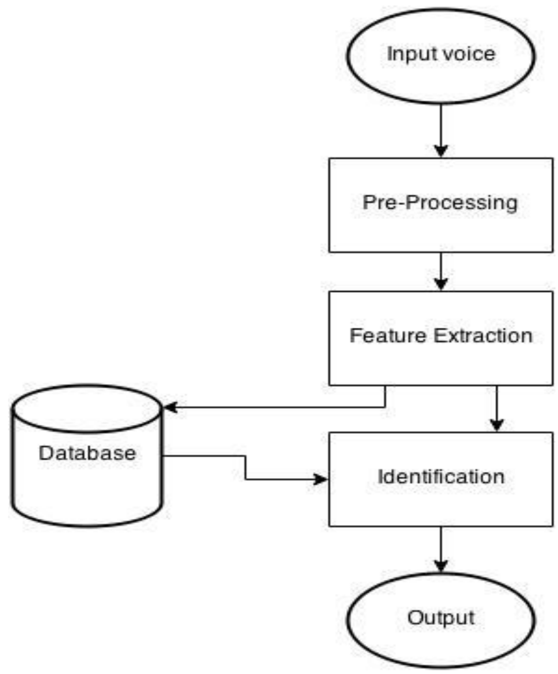

Figure (3.3) explains the system:

Figure3.2: Main steps of the system

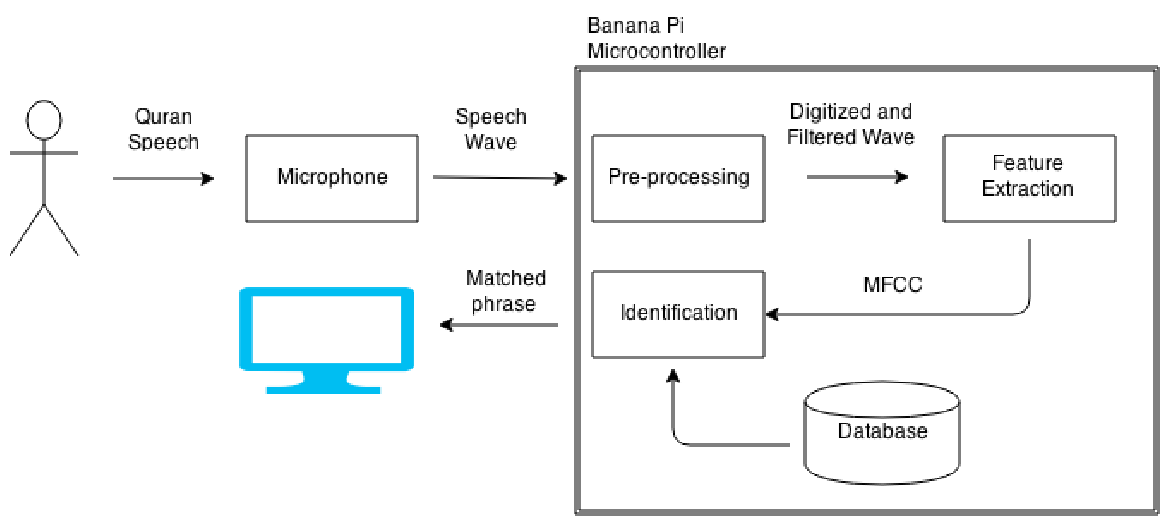

Figure3.3: Quran embedded system

Table 3.1 shows the whole system process steps:

Input: Quran speech *Output*: Result of *page of Quran* – notification for incorrect recitation. ***Stage 1: Building the template*** *Begin* Step 1: Input speech signal of Quran verse recitation is sampled. Step 2: Pre-emphasis is executed – Finite Impulse Response (FIR) filter. Step 3: The speech signal is framed. Step 4: Framed speech signal is windowed by using Hamming Window. Step 5: Fast Fourier Transform is applied to the windowed speech signal. Step 6: Mel-Frequency Cepstral Coefficients (MFCC) is calculated. Step7: DTW model is developed, i.e: λ *(Gd, Ld*) is evaluated and stored in the database. Or Step 7: HMM model is developed, i.e: λ *(A, pi0, mu*, *sigma*) is evaluated and stored in the database. *End*
Stage 2: Testing/Recognition *Begin* Step 1: Input speech signal of Quran verse. Step 2: Pre-emphasis is executed – Finite Impulse Response (FIR) filter. Step 3: The speech signal is framed. Step 4: Framed speech signal is windowed by using Hamming Window. Step 5: Fast Fourier Transform is applied to the windowed speech signal. Step 6: Mel frequency Cepstral Coefficients (MFCC) is calculated. Step 7: DTW model is developed, i.e: λ *(Gd, Ld*) is evaluated. Step 8: The DTW values compared with stored ones in the database by calculating Gd and Ld. Step 7: HMM model is developed, i.e: λ *(A, pi0, mu, sigma)* is evaluated. Step 8: The observation sequence and HMM values, obtained from the test input are compared with all models present in the database, through the Viterbi algorithm. Step 9: Quran accumulation filter is applied. Step 10: The recognition results – verses of the recognized word is decided based on the maximum value of log likelihood of the test data match with trained data. *End*

Table3.1: System process steps

3.2.1 Pre-processing:

Mainly, pre-processing is an essential stage in voice recognition to increase the accuracy and efficiency of the extraction process. Also, it's important to enhance the readability of speech processing. Here is how it goes:

3.2.1.1 Pre-emphasize filtering.

"*Pre-emphasis is considered as the first step of MFCC under the preprocessing stage in speech processing, which involved the signal conversion from analog to digital signal*"[8].

Speech signal may suffer from additive noise, and pre-emphasizing may kill this noise and get rid of it. Moreover, this step is considered as the first step before MFCC feature extraction.

The digitized sequence of samples x[n] stated as the relation below:

$$x[n] = x(nt)$$

Where: T is sampling period in (samples/ sec), and n is the number of samples.

First order high pass FIR (Finite Impulse Response) filter is used to do the pre-emphasizing.

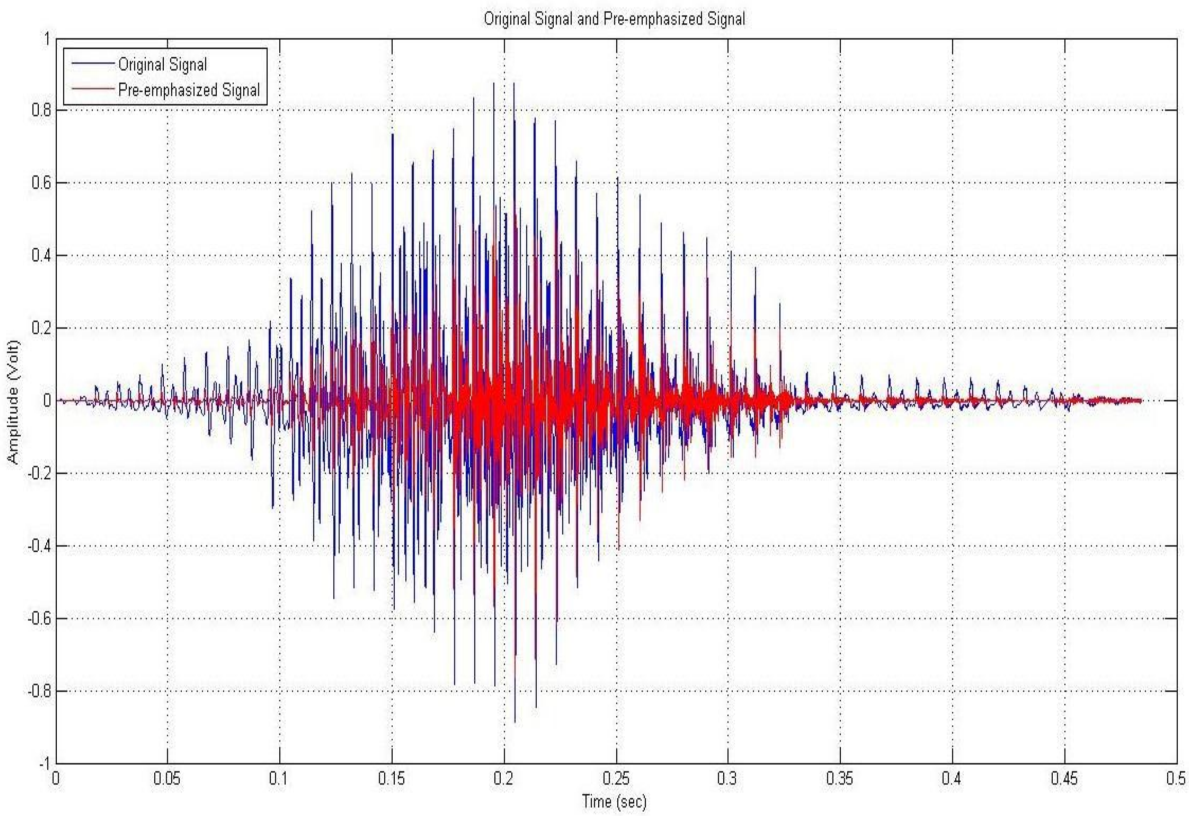

Figure3.2: Pre-emphasized signal [19]

3.2.2 Feature extraction

The feature extraction that will be done on the filtered signal consists of the following steps:

1. Framing: the process will be held on the input voice in terms of segments, so the signal will be split into several frames of small sizes. Moreover; since windowing is done in the next step where some data can be lost from the beginning or the end of each frame, an overlapping will be made between frames to reserve data from any loss.
2. Windowing: by minimizing the discontinuities at the beginning and at the end of each frame, the distortion in spectrum of the signal will also be minimized, and the common way to do that is by Hamming windowing [20] where 50% of overlapping to sufficient to cover the data loss[8].
3. Discrete Cosine Transform (DCT)
4. is applied on each frame, and the purpose is only because the convolution of signal in time domain is converted to multiplication in frequency domain, which is easier to process later, and it's done by using Fast Fourier Transform algorithm.
5. Mel Filterbank: to choose the target frequencies that contain info we are interested in, One of the ways is by using linear scale of frequencies, but the a better way is by using the scale that exists in human ear, which is Mel-scale that behaves as linear scale for frequencies under 1000Hz but, become logarithmic for frequencies above that.

   $$\text{Mel}(f) = 2595 * \log 10\,(1 + f/700)$$

   In order to do the filtering, a series of the *N* triangular band-pass filters is used for a filter bank whose center frequencies and bandwidths are selected according to the Mel-scale. As for choosing N; the increasing numbers of coefficients represent faster change in the estimated energies and thus have less information, so choose the first N (commonly 13) and the higher are discarded.

6. Converting the obtained Mel-frequencies back to time domain by Discreet Cosine Transform (DCT) to get the Mel-Frequency Cepstral Coefficients (MFCC) that will be the features of each signal frame.

3.2.3 Feature matching

In voice recognition, there're different design options with feature matching, the most famous one and people works with is Hidden Markov Model "HMM". On the other hand, Dynamic Time Warping "DTW" is another new efficient feature extraction technology. Moreover, both of these technologies should be implemented to work with the Quran Accumulation filter in the identification stage.

In this section, we'll cover these two feature extraction design options, and the accumulation filters.

3.2.3.1 Dynamic time wrapping "DTW"

Dynamic time warping is an algorithm for measuring similarities between two sequences of any data can be turned to a linear sequence which may vary in time or speed even if there we accelerations or de-accelerations. DTW could be used for graphics, audio, or data sequences [5].

"*The time alignment of different utterances is the core problem for distance measurement in speech recognition. A small shift leads to incorrect identification. Dynamic Time Warping is an efficient method to solve the time alignment problem. DTW algorithm aims at aligning two sequences of feature vectors by warping the time axis repetitively until an optimal match between the two sequences is found. This algorithm performs a piece wise linear mapping of the time axis to align both the signals.*"[20].

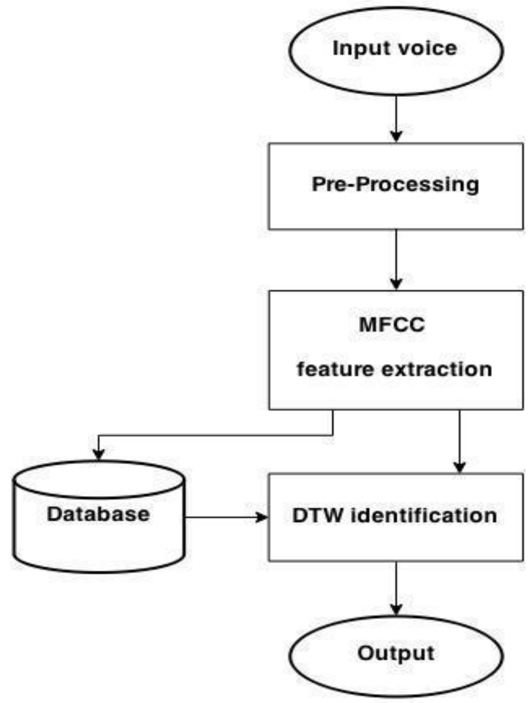

Figure3.5: System with DTW

DTW works as finding the local distance 'Euclidean distance' between cells of two sequences and then finding the global distance between the 2 aligned sequences to compare the corresponding elements based on this formula:

$$\mathrm{Dist}\,(x, y) = |x - y| = [\,(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2\,]^{1/2}$$

And the global distance is calculated through all possible routes of the grid of each 2 sequences

$$GD_{xy} = LD_{xy} + \min(GD_{x-1\,y-1}, GD_{x-1\,y}, GD_{x\,y-1})$$

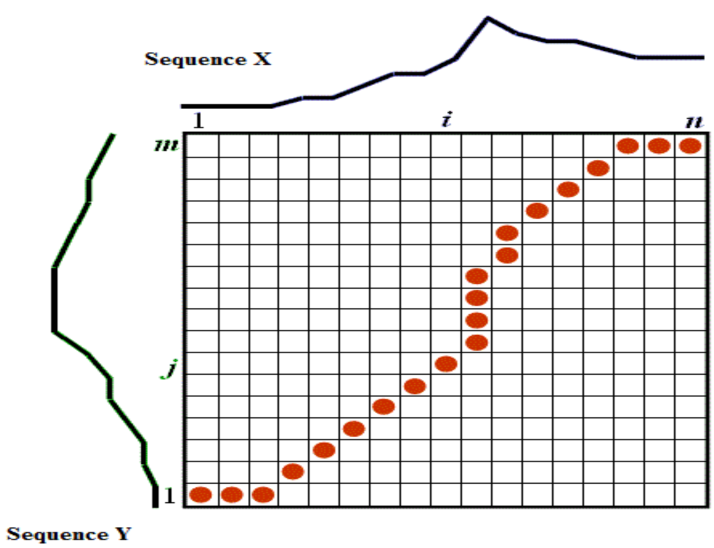

Figure3.6: Global distance grid [19]

3.2.3.2 Hidden Markov model "HMM"

For building the template using the HMM algorithm, there is an additional step after extracting the feature, which is the training where models are generated from the input speech, and the models themselves are stored in the database.

- Hidden Markov Model Training:

The training of Hidden Markov Model is used to model and represent the particular utterances of word or phoneme from the Quran recitation.
Each word or phoneme has a training set of k utterances -by different speakers- which represents the observation sequence of MFCC.
λ= (A, B, π)
Where:
A: the transition matrix,
B: observation matrix,
π: Initial state vector.

And the values obtained from the λ model for each set of feature vectors will be stored in the database.

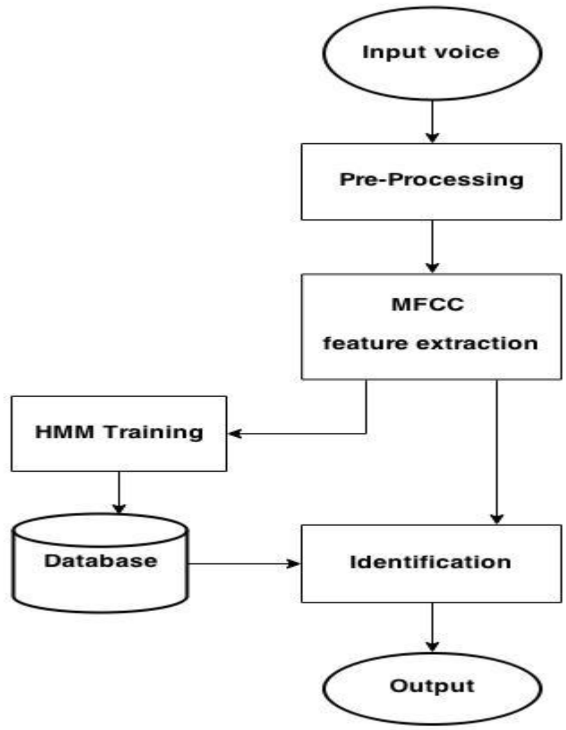

Figure3.7: System with HMM

- Hidden Markov model identification:

When we get the feature "O" to test from the input sample, the aim is to get the model$\lambda$from the database that maximizes the probability functionP(O/λ), where the proposed algorithm for the evaluation would be the "Viterbi Algorithm" [8] for HMM.

3.2.3.3 Accumulation filters

Kalman filter

It's an algorithm with a recursive nature uses sequences of measurements results that are observed over time to produce estimation tend to be more precise than an estimation based on a single measurement. It averages an uncertainty of a system state with a new measurement and returns a result which has the highest probability [22].

The algorithm works in a two-step process. In the prediction step, it produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty [23].

Quran accumulation filter

Here, in our project, Quran accumulation filter has been developed to solve the Quran recitation and tracking problem, and it will be used to accumulate results from HMM or DTW one after another to get rid of the error by accumulating results to get higher probability matching in real time processing while DTW or HMM works. It works as this: Once the outcome of the current speech frame (may be corrupted with some amount of error) is observed, it compares the results with the 2 previous results, and corrects it according to them if it was mistaken.

3.2.4 Implementation programming languages

With implementation there're several programming languages options, all of them work perfectly with the above design since they're powerful with mathematics functions.

3.2.4.1 Matlab language

MATLAB is a multi-paradigm numerical computing environment and fourth-generation programming language. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages [24].

3.2.4.2 Python language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together [25].

3.2.4.3 Scilab language

Scilab is an open source, cross-platform numerical computational package and a high-level, numerically oriented programming language. It can be used for signal processing, statistical analysis, image enhancement, fluid dynamic simulations, numerical optimization, and modeling, simulation of explicit and implicit dynamical systems and symbolic manipulations [26].

Chapter 4: Software

This chapter provides a description of the software used, some flowcharts, the algorithms used, and how the software developed drives the system.

4.1 System development part

In System Development part, Quran voice recognition system is developed due to extract, store and analyze the parameters of Quran recitation. The Mel-Frequency Cepstral Coefficient “MFCC” based algorithm is currently selected for feature extraction as it’s discussed clearly in the previous chapter and why it’s chosen. However, Dynamic Time Wrapping “DTW” based algorithm has been selected for feature matching - classification (comparison) since it’s a new algorithm for recognition and feature matching known that it’s simple, easy to implement in any programming language, produce the results quickly [5], and not classified as a “Black Box” algorithm like neural networks or HMM [20].Here, the process of speech recording (speech samples collection), features extraction, features training and pattern recognition formulate the Quran verse recitation recognition methodology.

4.2 Content development part

In the content development part, the samples of the Quran for 5 sourates; Kahf, Shu’ara, Qasas, Mo’mnon, and Taha, 8Verses of every sourate has been collected for the training stage by downloading them from an online Quran database for 7 Quran certified reciters and have been stored in the database folder for preprocessing and MFCC feature extraction, and the features vectors for the reciters have been stored in the database. However, with the testing stage, Quran samples of the same Verses and sourates have been recorded via a microphone for 2 different local certified reciters and have been stored in a different folder for testing. Moreover, all these Quran samples stored as (.wav) extension.

4.3 System flow chart

Embedded Quran voice recognition and tracking system flow chart emphasizes on the system’s flow of events. This engine has 5 main stages that include the pre-processing, features extraction and training, testing and recognition/classification, Quran accumulation filter, and displaying the results. Figure 4.1 shows those selection stages as well as process that probably occur for each stage.

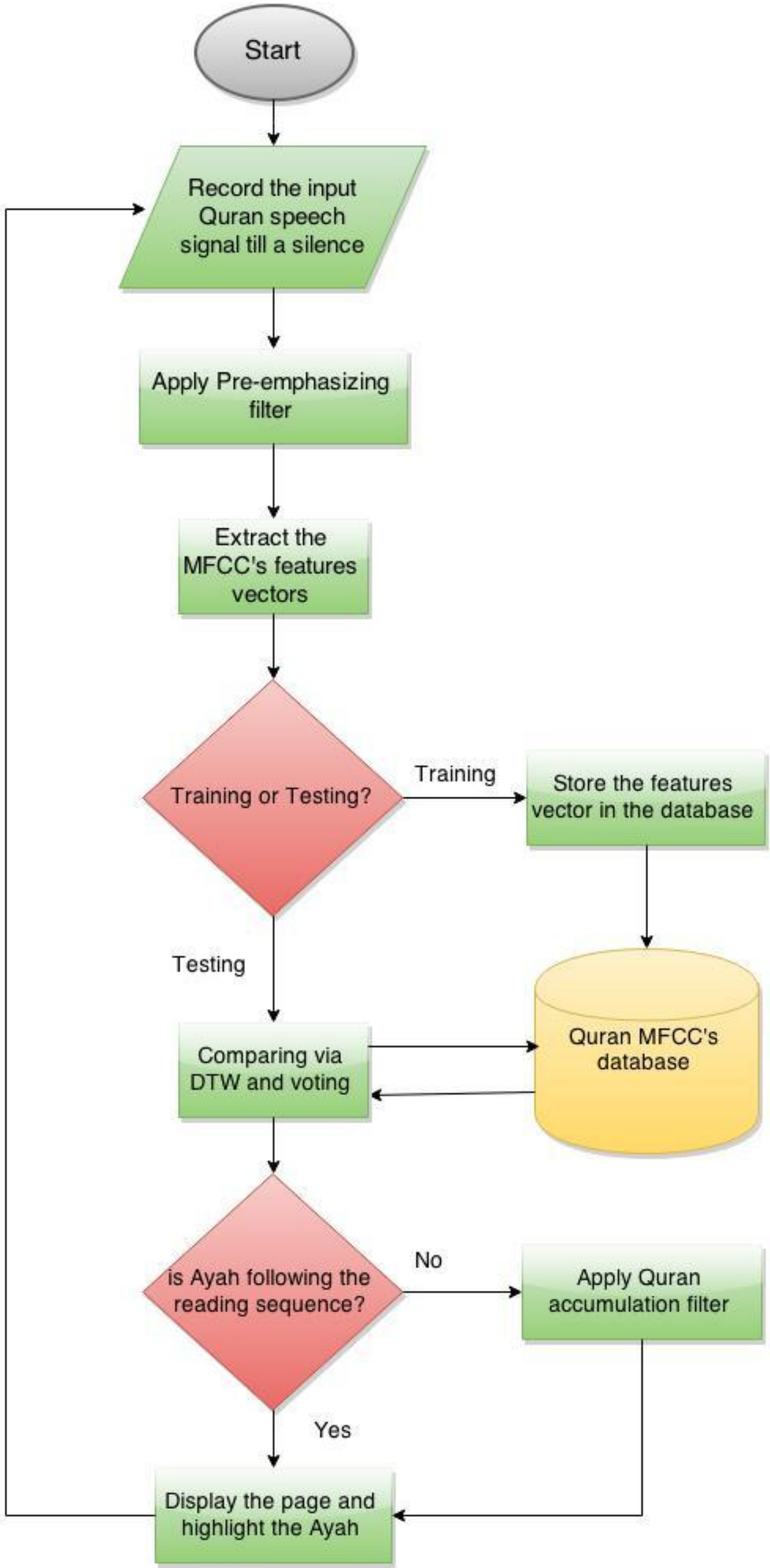

Figure4.1: System flow chart

Stage1:

Refer to figure 4.1, Quran speech signal will be recorded via the microphone till a silence, then it'll be pre-processed via a first order filter for smoothing, killing the noise, and echoing in the signal.

- Silence detection:

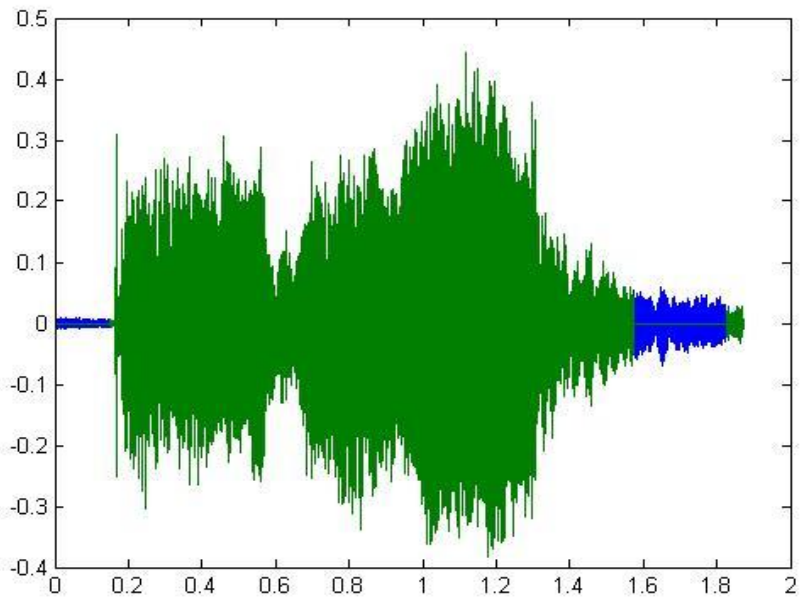

Figure 4.2: Silence detection

- Time domain

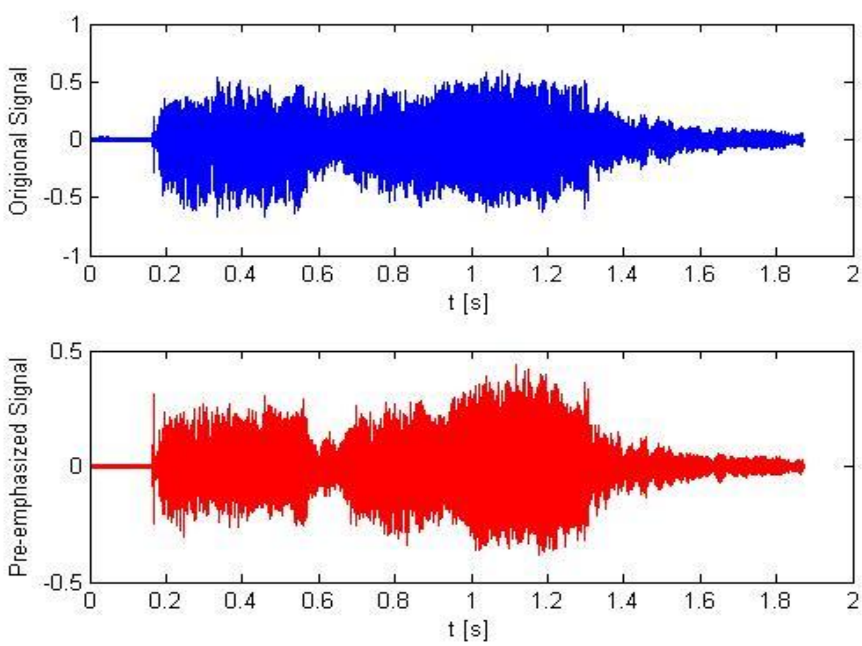

Figure 4.3: Signal in time domain

- Frequency domain

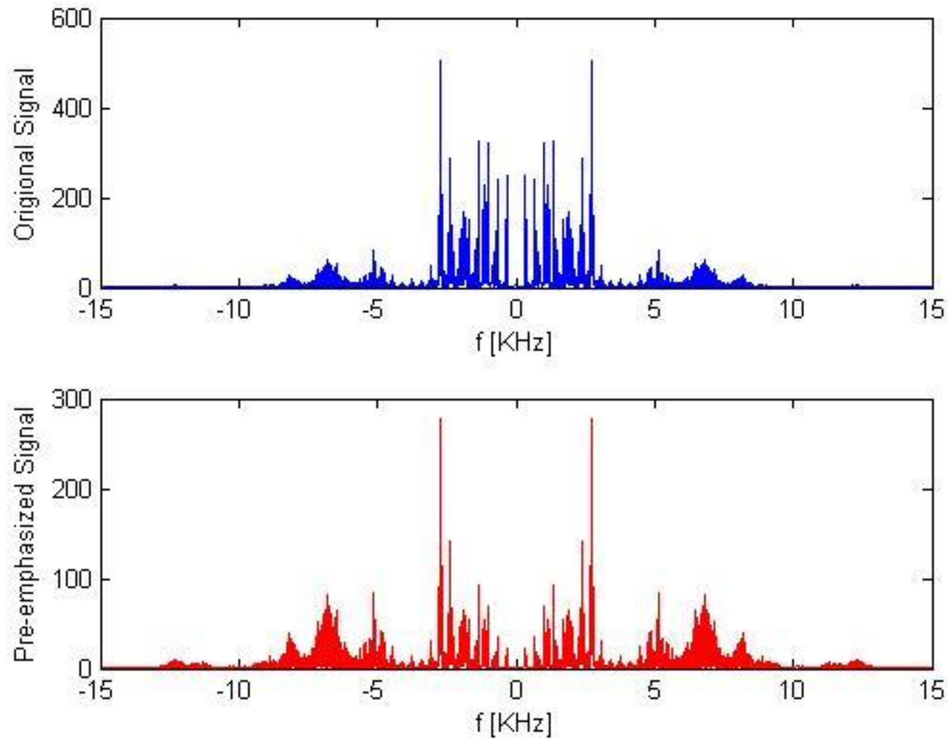

Figure 4.4: Signal in frequency domain

Stage2:

Feature extraction and training stage, the Quran signal will be framed and windowed before extraction the MFCC's features vector of the Quran samples in the database folder, and these vectors will be stored in the database for the speech recognition and feature matching stage.

Stage3:

```
Function Voting (TestVerse)
       For each Verse_i stored in DB
              Distances:= Find [Index_i, Distance_i] from DTW(TestVerse, Verse_i)
       EndFor
       For each Reciter_j in Distances
              Choose the nearest (min) distance with its index for Reciter_j
       EndFor
       If (index_i has max number of occurrences)
              Returnindex_i
       ElseIf (indices [i_1, i_2, i_3,.., i_n] have max number of occurrences)
              Returnindex_i that has the minimum sum of distances
End Function
```

In the testing stage classification will be done, a Quran recorded signal will be pre-processed, framed, windowed, and MFCC features vector will be extracted, and DTW applied as a feature matching- comparison algorithm, determining the nearest result for each reciter in the database, then voting will be applied for the results from the 7 different reciters to calculate the most common results as follows: the voting algorithm will calculate the occurrence for each result for different reciters, if there was a single repetition for the same Verse index then it'll be final result of the voting, however, if there was no repetition for any verse of the 7 reciters, the nearest answer which has the minimum distance will be the final result, also if there were equal number of occurrences, will calculate the summation of distances for every repetition, and the index of the minimum sum will be the final result of the voting.

Stage4:

```
Function QuranAccumulationFilter (TestVerse sequence)
	If (is in detecting stage)
		//Test the first three verses
		If (three in successive sequence)
			detection is correct
		ElseIf (two in successive sequence and one is out of the sequence)
			correct the wrong one and continue
		Else
			Fetch another test verse then detect the location of reading
		End
	ElseIf (is in tracking stage)
		If (new Test verse follows the previous)
			Continue
		ElseIf (two successive verses don't follow the previous)
		//error detected from reader
			beep //notification
		ElseIf (three verses don't follow the previous)
			//error detected from both reader and system
			beep //notification
End Function
```

In the this stage, Quran Accumulation filter will be applied, which is developed for this system to correct the mistaken results from the previous stage as follows: at first, the filter will compare the first three results to detect the exact sourate, page, and Verse by comparing each one of them with the other 2 results, the correct 3 Verses would be sequential with the indexing, so if any 2 of them weren't sequential the filter will correct it according to the other 2 verses, and if the there were no any 2 sequential verses, the filter will wait till the 4^{th} verse from the previous stage, and correct the previous according to them.

Then, tracking the Quran reading will follows by comparing every result from the previous stage with previous result from the filter. Such developed filter has a beep notification if any of the following occurs: if the Quran reader jumped to another verse in the same page or another page and read two sequential verses from there, or if the filter detects 3 non sequential results.

Stage5:

In the final stage, the system will display the resulted verse highlighted on a monitor, the display will start after the Quran reader reads the 3rd verse to give him an exact and true highlighted Verse, and then it'll keep displaying the verse supposed to be read while the Quran accumulation filter keeps tracking if any mistakes occurs.

4.4 Software programming language

As all of the testing and verification for the feature parameters and matching algorithms were done using Matlab, and since there's no Matlab version for microcontrollers yet; there were several options for choosing the implementation languages like: Python as one of the most common open source mathematical languages, but there was another choice, Scilab; an open source, cross platform mathematical language that has almost the same syntax as Matlab with few variances. The interfacing from Matlab to Scilab was done easily and there was a perfect match in the results, and as the code was installed and run successfully on the 'Banana Pi', Scilab has been confirmed as a feasible choice for such a problem.

4.5 Banana Pi implementation

The implementation on the Banana Pi has come through several stages. At first, Raspbian operating system has been chosen as the OS to run the Banana Pi since it's the highly compatible OS works with it. Secondly, we started converting the Matlab code to Scilab and making sure that there wasn't any syntax error during the converting by testing the functions one by one. Thirdly, checking the semantics of the Scilab code; in this stage we went through every function, checking its input and output arguments values, comparing them with the similar Matlab functions of the code. Finally, since there're some limited resources with the Banana Pi microcontroller such as stack size, RAM, and processing speed; functions optimizations were necessary and successfully done.Finally, the Banana Pi version with the Scilab showed the same output as Matlab.

- Training the database:

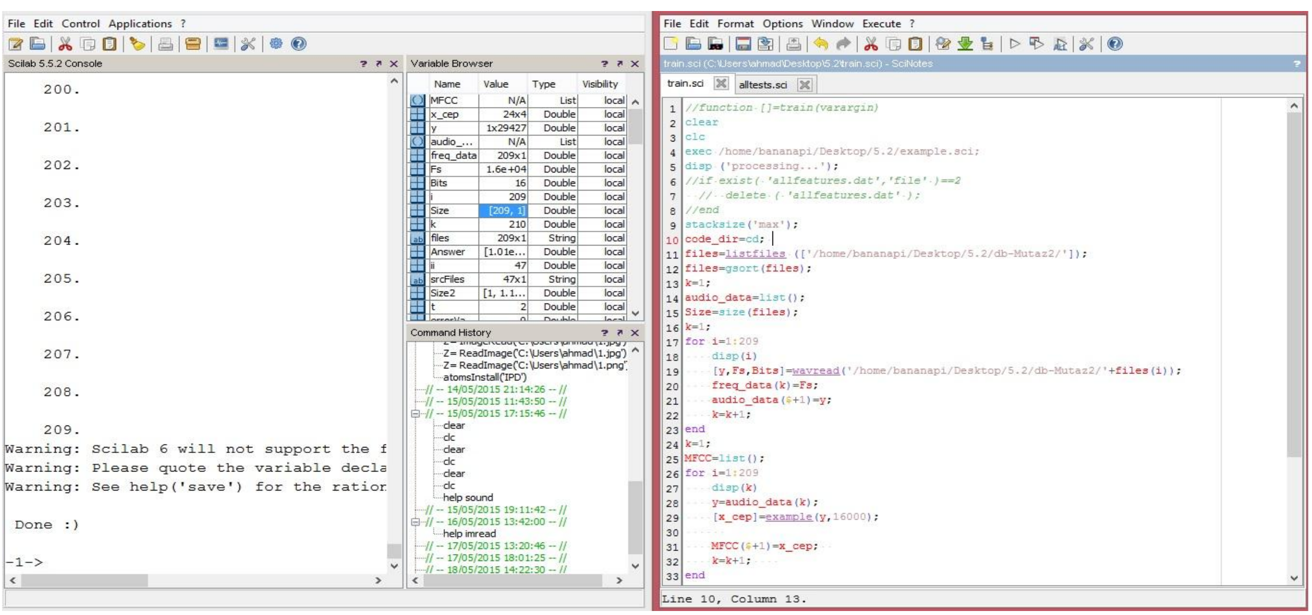

Figure 4.5: Training the database

- Running sample: 1st 8 verses from Al-Mo'mnon chapter.

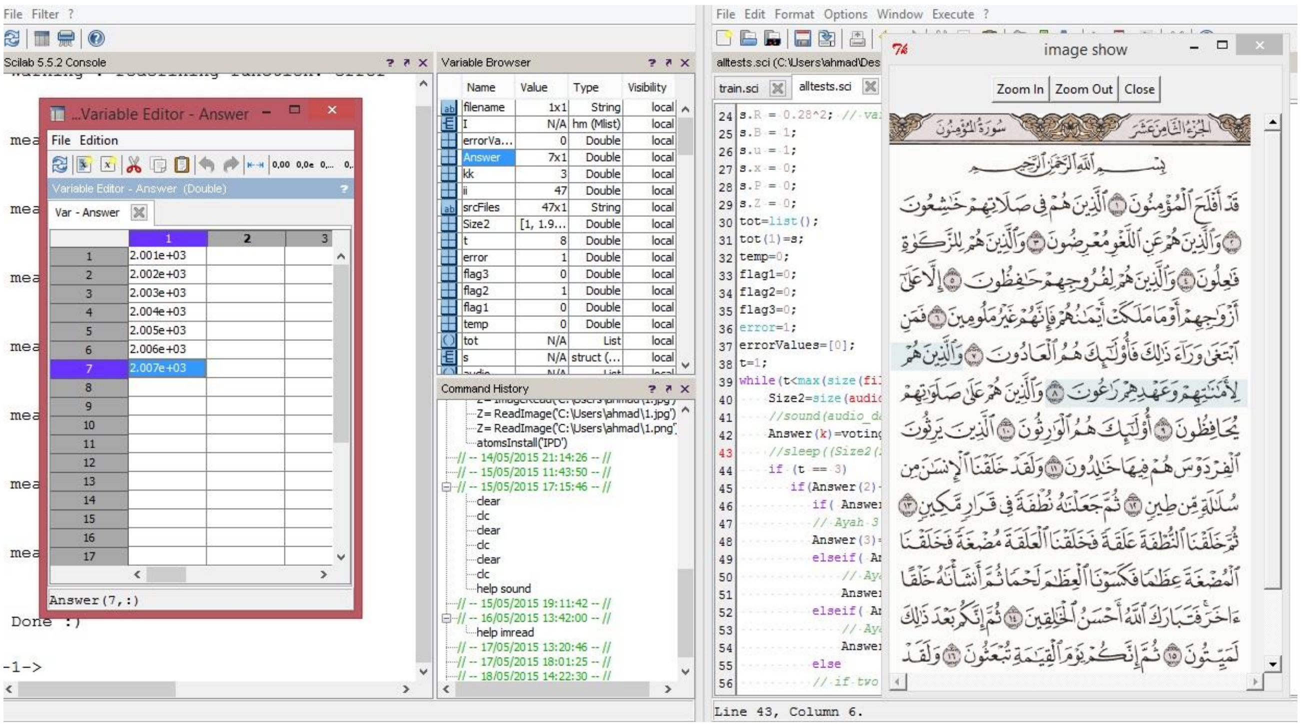

Figure 4.6: Testing 1st 8 verses from Mo'mnon chapter

Chapter 5: Validation and Discussion

This chapter contains experimental data and results as well as other extra information, analysis and discussion of the results obtained after the training and testing procedures executed on the system, also an evaluation of the performance of the overall system.

5.1 Feature extraction analysis

In such developed system for Quran speech recognition, there're five main parameters that affects the accuracy of the system and its results:

5.1.1 Frame length

In voice recognition and MFCC applications, most of them choose the frame length to be 20-25ms [8]. However, with our special case since we're dealing with a unique type of speech signal – Quran- which has special characteristics as explained in chapter 2, sections 2.2 and 2.3, we started doing the measurements from 5ms, increasing itby 5ms each time while keeping an eye on increasing the detection correctness for every verse. Surprisingly, with a frame length = 630ms, the system shows the best detection accuracy among frame lengths between 5ms and 830ms. With such a frame length we noticed that 630ms is the average length for reading a word in Quran.

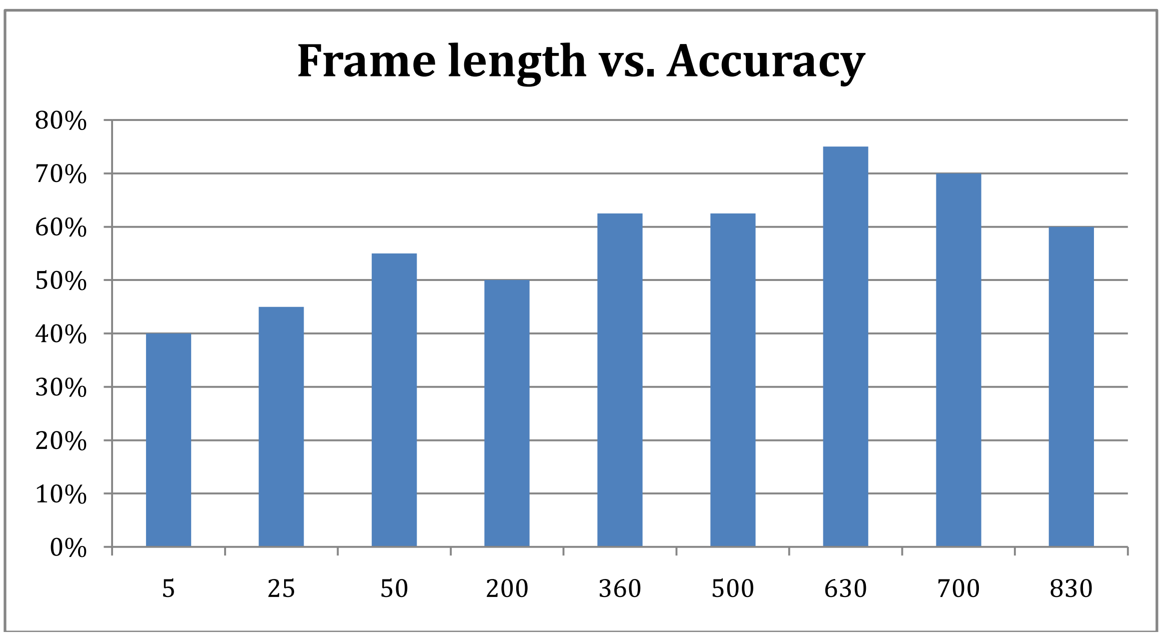

Figure5.1: Frame lengths and accuracy

5.1.2 Frame windowing and overlapping

By minimizing the discontinuities at the beginning and at the end of each frame, the distortion in spectrum of the signal will also be minimized [20], and the common way to do that is by Hamming windowingwith50% of overlapping between each frame and the next one.

- Wave before hamming

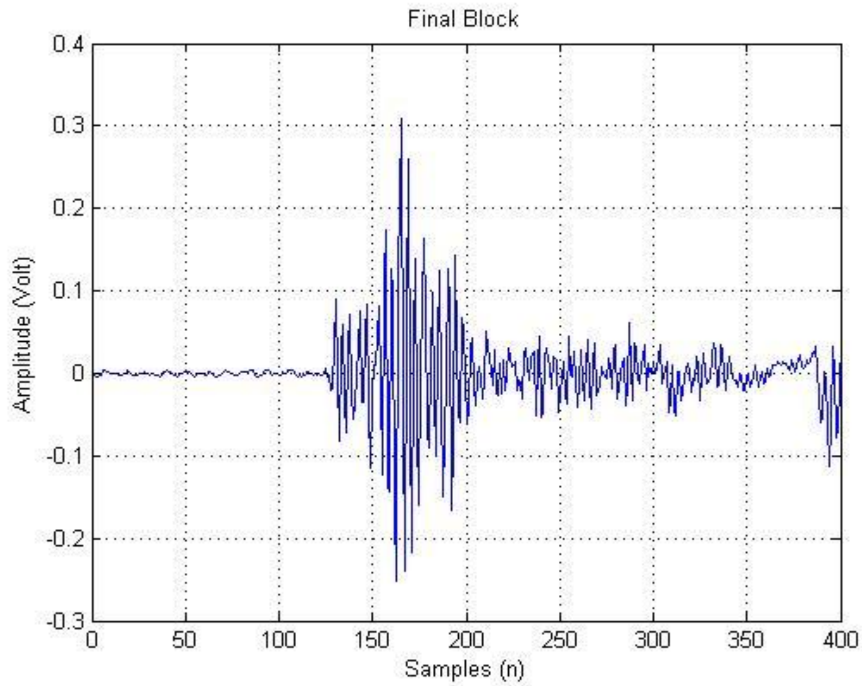

Figure 5.2: Signal before Hamming window

- Hamming window

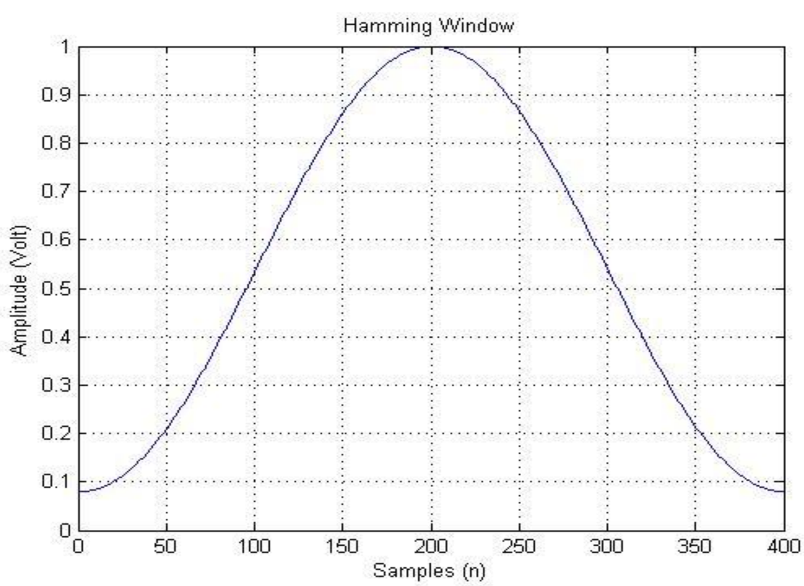

Figure 5.3: Hamming window

- Signal after hamming windowing:

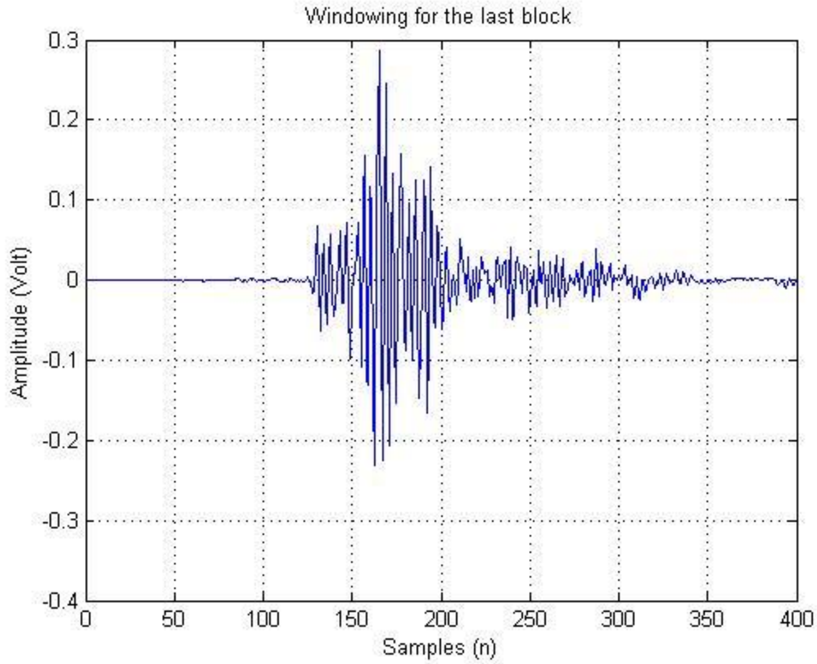

Figure 5.4: Signal after Hamming window

5.1.3 Number of filter banks

In voice recognition and MFCC applications, most of them choose initially 20 as the number of filter banks used for Mel frequency [20]. And since we didn't use more than the first 13 Cepstral coefficients, so there was no need to change this parameter.

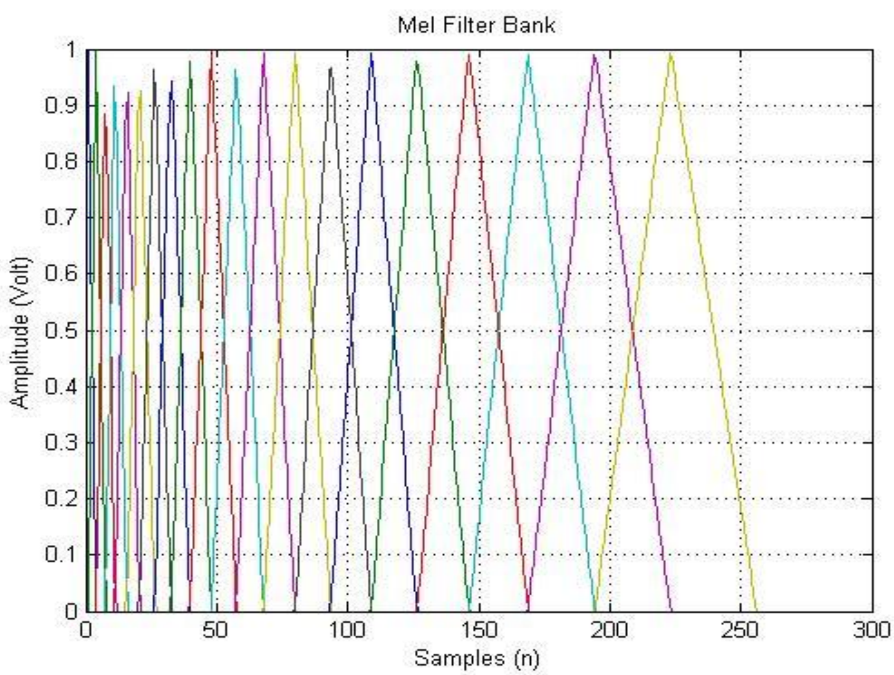

Figure 5.5: Mel filter banks

5.1.4 Number of features coefficients

With features coefficients, we obtain the first 13 Mel frequency Cepstral coefficients, and then getting rid of the first one since it represents the energy of the signal, which is not a unique value. Moreover, we had a problem with the variance in Quran speech signals lengths, as the short ones kept appear as results in every test for each verse; however, we came over this problem by taking the first and the second derivatives of the MFCC's representing the delta and acceleration for them. Practically, when we padded the acceleration to the MFCC vector the problem solved, and as a result the features vector for each frame consists of 24 coefficients.

In the table below, the MFCC and acc features vector for the 1st verse of Taha chapter:

Table 5.1: Features vector for 1st verse of Taha chapter

1	2.34968359648132	-1.41853275759685	-5.76973646065402	-7.14027971380390
2	5.66211671857988	-1.73443835224215	-5.40012658473562	-3.46719659104072
3	-18.9430867876737	-16.3007656714443	-11.8581748559274	-9.16181258637666
4	8.54737845354208	10.9716676484324	8.21742628343901	10.5774527328074
5	3.23789698755931	4.16544218589698	-0.0906773924931386	-2.23672010282773
6	-2.94396916978027	-13.1376097931299	-20.2775266409352	-19.1938177200342
7	-4.38847253604446	-1.27894967950273	1.94860123525735	5.41385259846303
8	-8.68379447259470	-0.342923426153180	9.37774309418740	9.89334172846126
9	-8.93350127972749	-4.95203591306825	-6.04094294889102	-4.47151130055453
10	4.85108342735481	7.25505296721839	9.42774971454275	5.36660494875165
11	5.91028437579174	3.12139778070898	2.18001226494431	1.01339068999253
12	13.3343806865115	13.6516950108806	18.2856522912655	13.4066953161619
13	-0.844527544789230	-0.222442217924506	0.530122958180719	0.976390719184750
14	-0.542425969704596	0.376250034666898	1.08673269952075	1.16941857007073
15	0.991771863472402	0.375201632314446	-0.458126216659399	-1.02731097104881
16	0.0845836766595086	0.00283783185600936	-0.105479732674843	-0.128573062724722
17	-0.701921677116769	-0.533843459386914	-0.0703526352902979	0.442980493683678
18	-1.19031691126612	0.242907380154087	1.54471943398917	1.95644268875609
19	0.960906374070384	0.386629303412934	-0.419272662200778	-0.974896384979460
20	1.63185901128569	0.0955643316677513	-1.49170784359420	-2.23020623068274
21	0.262870581045975	-0.0861302083905598	-0.407806854561736	-0.474557893739816
22	-0.0942688191562546	-0.526003609598949	-0.643446884575334	-0.406924249775582
23	-0.367685667516759	-0.0824282834582041	0.260874343581532	0.444162550426755
24	0.00201525674258716	-0.406794789347816	-0.551181870462808	-0.412576682375713

5.1.5 Quran speech frames and lengths

As mentioned before in this section, the final length for each frame is 630ms, and considering the 50% window overlap, the following table shows the relationship between Quran verses lengths and number of frames.

Table 5.2: Verses lengths and num. of frames

Verse	Length(sec)	#of frames	Verse	Length(sec)	#of frames
قَدْ أَفْلَحَ الْمُؤْمِنُونَ	5	16	طسم	10	31
الَّذِينَ هُمْ فِي صَلاتِهِمْ خاشِعُونَ	6	20	تِلْكَ آياتُ الْكِتابِ الْمُبِينِ	6	19
وَالَّذِينَ هُمْ عَنِ اللَّغْوِ مُعْرِضُونَ	6	18	نَتْلُوا عَلَيْكَ مِنْ نَبَإِ مُوسى وَفِرْعَوْنَ بِالْحَقِّ لِقَوْمٍ يُؤْمِنُونَ	13	41
وَالَّذِينَ هُمْ لِلزَّكاةِ فاعِلُونَ	6	19	طه	1	4
الْحَمْدُ لِلَّهِ الَّذِي أَنْزَلَ عَلى عَبْدِهِ الْكِتابَ وَلَمْ يَجْعَلْ لَهُ عِوَجاً	13	40	ما أَنْزَلْنا عَلَيْكَ الْقُرْآنَ لِتَشْقى	9	29
قَيِّماً لِيُنْذِرَ بَأْساً شَدِيداً مِنْ لَدُنْهُ وَيُبَشِّرَ الْمُؤْمِنِينَ الَّذِينَ يَعْمَلُونَ الصَّالِحاتِ أَنَّ لَهُمْ أَجْراً حَسَناً	15	48	إِلاَّ تَذْكِرَةً لِمَنْ يَخْشى	6	19
ماكِثِينَ فِيهِ أَبَداً	3	10	الرَّحْمنُ عَلَى الْعَرْشِ اسْتَوى	10	33

5.2 Classification analysis

5.2.1 Different reciters and database

Within the testing stage, we recorded Quran samples for 2 local certified reciters each of them of 8 sequential verses for 5 different sourates to check how much independent is our system. As noticed, reciter 1 accuracy's was 65%, and reciter 2 accuracy's was 68%. However, with enhancing the database; since the records initially chosen according to our intuition for clear records of certified reciters from the internet, the test cases of our samples showed considerable enhancement when adding one of the readers to the database then testing for the other: when we added reciter 1 Quran samples to the database it increased the detection accuracy to 75%, and when we added reciter 2 Quran samples to the database it increased the detection accuracy to 71%, which clearly shows the improvement happened in the database.

5.2.2 Dependent database

The Embedded Quran voice recognition system has been implemented to be used as an independent system for any Quran reciter; however, an observation must be considered regarding this fact. The system showed perfect accuracy to match two different records for the same reciter of the same verse, which means that it can function well for any reciter already save in the database.

5.3 Simulation vs. Implementation

As we used Matlab for simulation environment, and considering that we are dealing with huge amount of resources under such modern computers comparing with small microcomputers as Banana Pi, the analyses has three stages, taking into account the database and test samples as follows:

Database consists of 209 records of 8 verses from 5Sourates.

The test samples are first 8versesfrom Al-Mo’mnon chapter.

Table 5.3: Comparison between diff. OS’s and Implementation

	Training Time(sec)	Testing Time(sec)
Matlab–desktop version	141	33.5
Scilab –desktop version	112	30
Scilab –BananaPi version	495	182

Chapter 6: Conclusion and Future work

6.1 Conclusion

The main purpose of this project was to build an embedded system to help the Imam reading the Quran without mistakes in the aloud payers such Al-Taraweeh prayer, orto help Quran students to recite the Quran by an automated self-learning method, which will optimize the studying time and improve the learning process of the Quran.

The system has been developed by Matlab for simulation, and by Scilab programming language for as a real application based on a Banana Pi microcontroller connected with a microphone and a monitor. The microphone will collect the Quran speech, after that, the processing starts in the microcontroller by filtering the input speech, then recognizing the speech using Mel Frequency Cepstral Coefficients “MFCC” as feature extraction technology, and with Dynamic Time Warping “DTW” feature matching/classification technique for identification. The resulted Quran speech verse match from the Quran Accumulation filter- which is developed for this system- will be displayed highlighted on a monitor connected with the Banana Pi via an HDMI cable.

If the Quran reader made mistakes such jumping to another verse or page, the system will notify the reader for the error by a beep sound the reader can hear.

6.2 Future work

As a project have been developed for the first time of its kind, moreover; the result of this study solve the basic cases, many aspects can be enhanced for such a solution:

I. Database:

a. As mentioned in section 5.1.1, database optimization can be done using an automated algorithm (self organizing/learning) that store new records from users, chooses records having the best number of hits and get rid of records that do not match the inputs.

b. Section 5.1.2 discussion which is the dependency; can also be an advantage, that is even if the system made a wrong match in the first time, with more advanced tools, the system can correct its references and can easily functions with almost perfect accuracy with reciters records stored in the database, to make a correct decision in the second time and later on.

II. Quran accumulation algorithm:

The current algorithm deals with only basic cases, but it can also be enhanced to deal with many other cases including the repetitions of the same verses or stopping in the middle of verses then reading it from the beginning again, one step backward then continuing reading and many other scenarios.

III. Detection scale:

Increase the detection accuracy to deal not only whole verses, but also in terms of words, so that the system would be able to detect even missing words.

References

[1] Awais, M.M.et al. (2003) '*Simulation of Arabic Phoneme Identification through Spectrographic Analysis'*, Department ofComputer Science, University of Engineering & Technology, Lahore Pakistan.
[2] Habash, M. (1986) '*How to memorize the Quran'*, Dar al-Khayr, Beirut.
[3] Hansen, J.C. (2003) '*Modulation based parameter for Automatic Speech Recognition*',Master Thesis of Department of Electrical Engineering, University of Rhode Island USA.
[4]Mohd, Y.I. et al. (2008) '*Quranic Verse Recitation Recognition Module for Support in j-QAF Learning*'Faculty of Science Computer and Information Technology, University of Malaya.
[5]Vergyri, D.et al. (2004) '*Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition'* COLING Workshop on Arabic-script Based Languages, Geneva.
[6]Stolcke, A. et al. (2004) ' *Morphology-based language modeling for conversational Arabic speech recognition'* Eighth International Conference on Spoken Language ISCA.
[7]Kirchhoff, K. (2001) '*Novel Speech Recognition Models for Arabic'* Johns-Hopkins University Summer Research Workshop.
[8]Ibrahim, N.(2010) '*Automated Tajweed Checking Rules Engine ForQuranic Verse Recitation*' Faculty of Computer Science and Information Technology
University Of Malaya Kuala Lumpur.
[9]Ahmad, A.M. et al. (2004) 'Recurrent Neural Network with Backpropagation through Time for Speech Recognition' *IEEE International Symposium on Communications & Information Technology*, ISCIT '04.Volume 1, pp. 98 – 102.
[10]Bashir, M.S. et al. (2003) '*Simulation of Arabic Phoneme Identification through Spectrographic Analysis*' Department of Computer Science LUMS, Lahore Pakistan.
[11] Ehab, M. et al.(2007) *'Speaker Independent Quranic Recognizer Based on Maximum Likelihood Linear Regression*' Proceedings of World Academy of Science, Engineering and Technology Volume 20.
[12] Chetouani, M. et al.(2002)2002, '*Neural Predictive Coding for speech Discriminant Feature Extraction: The DFE-NPC*', *ESANN'2002 Proceedings* – European Symposium on Artificial Neural Network, Bruges, Belgium, pp. 275-280.
[13] Rabiner, L. Rabiner, L.R., (1989) '*A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition*', *Proceeding of the IEEE*, Volume 7, No.2.
[14] Nathan, K. et al. (1995).'On-line Unconstrained Handwriting Recognition Based On Probabilistic Techniques'.IBMWaston research center, Now York USA.
[15] McCulloch, W. (1943). '*A Logical Calculus of Ideas Immanent in Nervous Activity'*, Bulletin of mathematical biophysics, *vol.5, p.115-133.*
[16] Muller, M., Information Retrieval for Music and Motion, Ch. 4 2007.
[17] Forney, G.D. (2005) '*The Viterbi Algorithm: A Personal History'* Cambridge, MA USA.
[18] http://www.bananapi.org/p/product.html-Accessed in 15th-Nov-2014.
[19] Geeta,N. et al. (2013) '*Isolated Speech Recognition Using MFCC and DTW'* International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 8.

[20] Becchetti,C. et al. (1999) 'Speech Recognition: Theory and C++ Implementation', England.
[21]Karpov,E. (2003)*'Real Time Speaker Identification'*, Master`s thesis, Department of Computer Science, University of Joensuu.
[22] Einicke, G.A. (2012) '*Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*', Rijeka, Croatia: Intech. ISBN 978-953-307-752-9.
[23]Chen, G. et al. (2009) '*Kalman Filtering with Real-Time Applications*', Springer Series in Information Sciences **17** (4th ed.). New York: Springer. p. 229. ISBN 978-3-540-87848-3.
[24] "The L-Shaped Membrane". MathWorks. 2003. Retrieved 7 February 2014.
[25] https://www.python.org/doc/essays/blurb/, visited on May 18th, 2015
[26] Thomas Trappenberg (2010). *Fundamentals of Computational Neuroscience*.Oxford University Press. p. 361. ISBN 978-0-19-956841-3.