STATUS OF THESIS

Title of thesis | SPEECH RECOGNITION FOR CONNECTED WORD USING CEPSTRAL AND DYNAMIC TIME WARPING ALGORITHMS

I LINDASALWA BINTI MUDA

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

_____

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

_____

Endorsed by

Signature of Author

Permanent address:
127 B Kampung Buluh,
21700 Kuala Berang,Terengganu
Terengganu Darul Iman

Date: _____

Signature of Supervisor

_____
_____
_____

Date: _____

UNIVERSITI TEKNOLOGI PETRONAS

"SPEECH RECOGNITION FOR CONNECTED WORD USING CEPSTRAL AND DYNAMIC TIME WARPING ALGORITHMS"

by

LINDASALWA BINTI MUDA

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree of Master of Science in Electrical and Electronics Engineering.

Signature: _____

Main Supervisor: Assoc.Prof. Dr. Irraivan Elamvazuthi

Signature: _____

Head of Department: Assoc.Prof .Dr. Rosdiazli bin Ibrahim

Date: _____

UNIVERSITI TEKNOLOGI PETRONAS

SPEECH RECOGNITION FOR CONNECTED WORD USING CEPSTRAL AND DYNAMIC TIME WARPING ALGORITHMS

by

LINDASALWA BINTI MUDA

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL & ELECTRONICS ENGINEERING

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SRI ISKANDAR

PERAK

SEPTEMBER 2014

DECLARATION OF THESIS

Title of thesis | SPEECH RECOGNITION FOR CONNECTED WORD USING CEPSTRAL AND DYNAMIC TIME WARPING ALGORITHMS

I LINDASALWA BINTI MUDA hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author

Permanent address:
127 b Kampung Buluh,
21700 Kuala Berang,Terengganu
Terengganu Darul Iman

Date: _____

Signature of Supervisor

_____
_____
_____

Date: _____

ACKNOWLEDGEMENT

Many people have contributed to this thesis. First and foremost, everything in me that I count that as good comes only from the Allah s.w.t, who leads, grants and guided me along in completing this thesis. All honor and glory be to Him.

I deeply thank my supervisor, AP. Dr. Irraivan Elamvazuthi for his guidance. Thanks for sharing invaluable advances and experiences. His attitude and enthusiasm in doing research and his consistent vision to make higher technology research grounded in reality, applicable and even commercialized always inspired me. Secondly, very special thanks to my co-supervisor, Dr. Mumtaj Begam for her endless support to build up my foundation knowledge on Speech Recognition field.

I am also indeed indebted to all the friends for their generosity in sharing the resources and knowledge with me during this research. Special thanks to UTP lecturers, staff and other talented individuals for providing me resources, and conducive environment to make my research a success.

Lastly, I dedicate this thesis to my beloved husband, daughter and son. Not forgetting, I would also extend my sincere gratitude and thanks to all my family members for their encouragement and support.

Lindasalwa Binti Muda

SEPTEMBER 2014

ABSTRACT

Speech Recognition or Speech Recognizer (SR) has become an important tool for people with physical disabilities when handling Home Automation (HA) appliances. This technology is expected to improve the daily life of the elderly and the disabled so that they are always in control over their lives, and continue to live independently, to learn and stay involved in social life. The goal of the research is to solve the constraints of current Malay SR that is still in its infancy stage where there is limited research in Malay words, especially for HA applications. Since, most of the previous works were confined to wired microphone; this limitation of using wireless microphone type makes it an important area of the research. Research was carried out to develop SR word model for five (5) Malay words and five (5) English words as commands to activate and deactivate home appliances. The research involved the investigation of applying suitable algorithms to handle 'connected words' based on wired and wireless microphones for both English and Malay words. The combination of non-parametric method for modelling the human auditory perception system, Mel Frequency Cepstral Coefficients (MFCC), and speech enhancement process to reduce microphone effect, Cepstral Means Subtraction (CMS) known as MFCCCMS was investigated as an extraction algorithm. The non-linear sequence alignment known as Dynamic Time Warping (DTW) and Vector Quantization (VQ) have been used as feature matching algorithms. The research findings show that the 'word accuracy' for 'connected Malay words' for the proposed algorithms are from 85.3% to 99.7% using wireless microphone. Similar results were obtained for 'connected English words' where the 'word accuracy' (WA) was found to be between 89.3% and 99.3%.

ABSTRAK

Sistem Pengecaman Suara merupakan satu alat yang penting bagi orang kurang upaya (OKU) fizikal apabila mengendalikan peralatan automasi rumah. Teknologi ini dijangka dapat memperbaiki kehidupan harian warga tua dan OKU supaya mereka sentiasa dapat mengawal kehidupan dan terus hidup berdikari, belajar dan melibatkan diri di dalam kehidupan sosial. Matlamat tesis ini bertujuan menyelesaikan kekangan sistem pengecaman suara untuk perkataan Melayu dimana ianya masih di peringkat awal. Selain itu, penyelidikan untuk perkataan Melayu sangat terhad terutamanya untuk penggunaan peralatan automasi rumah. Oleh kerana, kebanyakan kerja kajian sebelum ini terhad kepada mikrofon berwayar, penggunaan jenis mikrofon tanpa wayar menjadi satu bidang yang penting untuk dikaji. Kajian ini dijalankan untuk membangunkan model pengecaman perkataan untuk lima (5) arahan perkataan bersambung Melayu dan lima (5) arahan perkataan bersambung Bahasa Inggeris untuk mengaktifkan dan menyahaktifkan peralatan automasi rumah menggunakan sistem pengecaman suara.. Kajian ini melibatkan penyiasatan algoritma yang sesuai untuk mengendalikan arahan berdasarkan mikrofon berwayar dan tanpa wayar untuk kedua-dua perkataan bersambung Inggeris dan bersambung Melayu. Gabungan kaedah bukan parametrik untuk sistem model auditori manusia, Mel Frequancy Cepstral Coefficient (MFCC) dan proses peningkatan ucapan untuk mengurangkan kesan mikrofon Cepstral Mean Subtraction (CMS), telah disiasat sebagai algoritma pengekstrakan dipanggil MFCCCMS. Teknik bukan penjajaran urutan linear dikenali sebagai Dynamic Time Warping (DTW) dan Vektor Pengkuantuman (VQ) telah digunakan sebagai algoritma padanan. Kajian menunjukkan bahawa ‘ketepatan perkataan’ untuk perkataan bersambung Melayu adalah 85.3% sehingga 99.7% menggunakan mikrofon tanpa wayar. ketepatan perkataan bagi perkataan bersambung Inggeris adalah 89.3% hingga 99.3%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OFABBREVIATIONS

AI	Artificial Inteligence
AP	Acoustic Phonetic
AMDF	Average Magnitude Different Function
ANN	Artificial Neural Network
ANFIS	Adaptive Neuro Fuzzy Inference System
ASR	Automatic Speech Recognition
ADC	Analog to Digital Converter
BWMFCC	Bark Wavelet Mel Frequncy Cepstral Coefficients
BPNN	Back Progation Neural Network
CMS	Cepstral Means Subtraction
CRF	Conditional Random Field
CMN	Cepstral Mean Normalization
CWRTs	Crosswords Reference Template and DTW
DARPA	Defense Advanced Research Project Agency
DDA	Driven Decodong Algorithm
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
FT	Fourier Transform
FE	Feature Extraction
FFT	Fast Fourier Transform
FM	Feature Matching
f_s	Sampling Frequency
FM	Frequency Modulation
GA	Genetic Algorithm
HA	Home Automation
HMM	Hidden Markov Model
HTK	HMM Toolkit
LBG	Linde-Buzo-Gray

LHEQ	Linear Histrogram Equalization
LPC	Linear Predictive Coefficient
LPCC	Linear Predective Spectral Coefficients
LVCSR	Large Vocabulary Continous Speech Recognition
MFCC	Mel Frequncy Cepstral Coefficients
MFCCCMS	Mel Frequency Cepstral Means Subtraction coefficients
MLP	Multilayer Linear Perceptron
MPE	Minimum Phone Error
PR	Pattern Recognition
PT	Pattern Training
PM	Pattern Matching
PCM	Pulse Coded Modulation
PLP	Perceptron Linear Predictive
RASTA	Relative Spectra
SD	Speaker Dependent
SI	Speaker Independent
STE	Short Time Energy
SR	Speech Recognition
SIEM	Speech Interactivity Embedded Module
T1	Template 1
T2	Template 2
UTP	University Teknology Petronas
V/NV	Voice/No-voice Detection
VAD	Voice Activity Detection
VQ	Vector Quantization
ZCR	Zero Crossing Rate
WA	‘Word Accuracy’
WER	‘Word Error Rate’

CHAPTER 1

INTRODUCTION

This chapter begins by proving a background of Speech Recognition (SR), problem statement, research objectives, contribution of the thesis and finally, the outline of the thesis.

1.1 Introduction

An understanding of the goals and methods of the past can help us to envision the advances of the future [1]. The SR from 1920s until now has grown rapidly with different techniques, algorithms and methods. SR also known as Automatic Speech Recognition (ASR) is the ability of the machine or computer to recognize the spoken word. The output from this ASR can be letters, words or phrases. Today, ASR technology is nearly ubiquitous and available commercially.

The Home Automation (HA), also known as Smart Home, refers to the use of computer and information technology to control home appliances and features [2]. It is automation of the household activities. The HA system is intended to activate and deactivate all home appliances and automatically control things around home using speech command. HA of early prototypes of commercial products were developed at the end of 1970s [3]. The recent development in technology permits the use of radio frequency technology in wireless mode, capable of communicating with different appliances and amongst each other. For those with limited movement such as the elderly and people with physical disabilities who are living by themselves, there is a growing need for HA system that can make their lives easier without the use of wired communication.

Several SRs for HA systems have been presented by many researchers [4-7]. All researchers have shown that the development of HA system would be profitable and adaptable to use not only for both elderly and disabled person; it is also capable of making a pleasant and very comfortable for home users. Numerous research have

been carried out on SR with different kinds of signal modeling approaches such as Pattern Recognition (PR), Artificial Intelligence (AI) and Acoustic Phonetic (AP) for various languages such as English, Malay, French, Chinese, etc. [8-13]. According to the literature, the AP approach is the oldest SR approach since 1950; however, the PR is commonly used because it is simple, and easier to understand the mathematical computation and theoretical justification used in the training and matching processes. On the other hand, the AI is the youngest, and the least-known approach is the combined PR and AP approach [14-16]. Many techniques have been developed to improve the performance of SR for SD and Speaker Independent (SI) methods. It is widely accepted that an effective algorithm is able to provide at least 10% error reduction [17, 18].

The literature review in this research focused on SR for general and HA applications using all the three (3) approaches; namely, PR, AI and AP. The general application that uses SR technology are digit recognition, desktop application, robotic control, wheelchair control, automated dictation, other language dependent systems and others.

1.2 Problem Statement

The current Malay word for SR is limited by following constraints:

1. Limitations of ‘word-based models’

   In general applications, only a few researchers have used ‘isolated English words’ and ‘isolated Malay words’, and none in ‘connected English words’ and ‘connected Malay words’. Under HA, theres isn’t any use of Malay words.

2. Lack of Malay words accuracy

   Amongst all the three approaches (PR, AI and AP), PR produced the best ‘word accuracy’ for general and home automation applications. For many languages, the ‘word accuracy’ was between 68% to 100%, whereas, it was about 80.5% for Malay words. Therefore, there is a need to improve the ‘word accuracy’ of Malay words using different algorithms to be on par with English words.

3. Speech input device or microphone type

   It was found that for general and HA applications, most of the work is confined to wired microphone. Only a few researchers have used wireless microphone for HA, and none for general applications. Wireless speech input device is important for those with limited movement such as the elderly and people with physical disabilities who are living by themselves in a HA environment. The limited use of wireless in HA makes it an important area for research in SR.

1.3 Research Objectives

In view of the foregoing problems, the main objective of this research is to develop and evaluate SR algorithms to improve the performance in terms of 'word accuracy' for 'connected Malay words' and ' connected English words' for HA environments using wired and wireless speech input device (microphone). Specifically, the objectives of this research are as the following:

1. To develop algorithms to enhance the SR performance for 'connected Malay words' and 'connected English words'
2. To evaluate the performance of wired and wireless speech input devices for SR
3. To assess and validate the performance of the proposed SR algorithms

1.4 Scope of Research

Thus, the scope of the research is based on the following:

1. The SR experiments were based on Speaker Dependent (SD) type.
2. All the experiments used speech command of five (5) 'connected Malay words' and five (5) 'connected English words' uttered by fifteen (15) speakers.
3. All the experiments used mono wired and wireless stereo headsets as speech input devices.

4. The Mel Frequency Cepstral Coefficients (MFCC) and Cepstral Mean Subtraction (CMS) algorithms were used as Feature Extraction (FE) techniques.
5. The Dynamic Time Warping (DTW) and Vector Quantization (VQ) algorithms based PR approach were used as Feature Matching (FM) techniques.
6. The combination of different features for extraction and matching were evaluated and accuracy is determined.

1.5 Contribution of Thesis

Overall, four (4) techniques were developed where the first technique uses MFCCCMS and DTW as FE and FM algorithms respectively. The second technique involves the use of MFCC and DTW as FE and FM algorithms respectively. The third technique uses MFCCCMS and VQ as FE and FM algorithms respectively. The fourth technique involves the use of MFCC and VQ as FE and FM algorithms respectively. From this point onwards, all techniques are called MFCCCMS+DTW, MFCC+DTW, MFCCCMS+VQ and MFCC+VQ. The contributions of this research are summarised as the following:

1. Four algorithms, namely MFCCCMS+DTW, MFCC+DTW, MFCCCMS+VQ and MFCC+VQ were developed for ‘connected Malay words’ where it has not been reported in prior literature to the best of the author’s knowledge.
2. The performance proposed by the algorithms of MFCCCMS+DTW, MFCC+DTW, MFCCCMS+VQ and MFCC+VQ for ‘connected Malay words’ has been enhanced over 90%. The algorithms were able to reduce the error not more than 10% with accuracy of 98.7%, 98.3%, 99.3% and 100% respectively using wired, and %99.7%, 98.3%,99.3% and 98% respetively using wireless.
3. This research also proposes that the ability presented by MFCCCMS to extract features, and DTW and VQ to compare the test pattern in addition to enhancing the accuracy of ‘conncected Malay words’ could be a viable solution for SR in HA applications.

1.6 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 provides the background and literature review of the relevant topics. In this chapter, several aspects of SR, such as history of SR, application of SR in general, HA for people with disabilities are briefly reviewed. Finally, critical analyses of the literature are provided.

Chapter 3 presents five (5) major elements of SR Engines .This is followed by discussion on Digital Signal Processing (DSP) techniques for Feature Extraction and Feature Matching. Finally, it discusses the experimental procedure of data collection and summarizes all the topics of the chapter.

In Chapter 4, the comprehensive results and discussion is provided for General and HA apllications for both ‘connectd English words’ and ‘connected Malay words’. Finally, experimental results demonstrate the effectiveness of the proposed approach. Chapter 5 concludes the thesis and provides a general discussion and ideas for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter presents a literature review of speech recognition (SR) field. It begins with the fundamentals of SR, followed by historical timeline and parameters of SR. Under the parameters, elements such speakers, utterances, size of vocabulary and recordings are discussed. Subsequently, related works of SR in general and HA applications are provided. Then, critical analysis of literature review is presented and finally summary is given.

2.1 Fundamental of Speech Recognition

The sound that is commonly known as speech begins with lungs contracting to expel air, which carries sound of an approximately Gaussian frequency distribution [19]. The main vocal organs are the lungs, larynx, pharynx, nose and mounth [20]. Figure 2.1 shows the human vocal organ that is responsible for speech production.

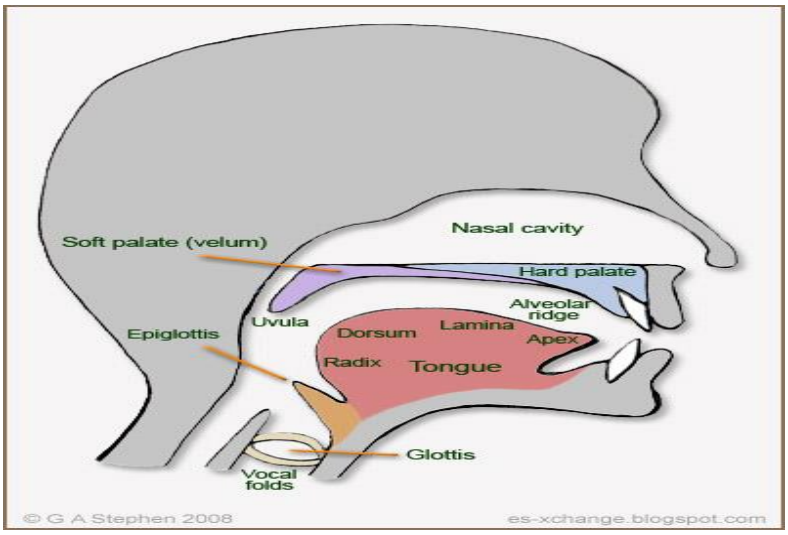

Figure 2.1: Sectional diagram of human vocal apparatus [20]

The process begins when the air is forced up through the bronchial tract past a set of muscle folds at vocal chords and the tensed vocal cord with larynx will cause vibrating. The air then enters the rear of the mouth cavity where it follows one of two paths. The first one is over around tongue, past the teeth and out through the mouth. The second route is through the nasal cavity and this is the only route when the velum is closed [19]. Technically, while speech sound is produced, the air flow from lungs passes the glottis, and then, go through throat and mouth. The speech signal and all its characteristic can be presented in time or frequency domain. According to researchers [21 - 22], the speech can be excited in three possible ways to classify event in speech. The ways are, Silence (S), where the glottis is closed and no speech is produced. The second way is, Unvoiced (U), the glottis is open and the air passes a narrow passage in the throat or mouth to make vocal cords not to vibrate. This results in a random speech waveform. Thirdly, Voiced (V), is a way which a closure in the throat or mouth will raise the air pressure suddenly making the vocal cords to be tensed and thus, creating a vibrating effect periodically.

By definition, speech is the vocalized form of human communication and it is the syntatic combination of lexicals and names which are from very large vocabularies [23]. Speech is used as natural mode for communication amongst people. It comes naturally during early childhood, without instruction and human beings continue to use it throughout their lives without realizing the uniqueness of each speech signal.

Since the earliest days of computing, there have been various research to understand human speech for development of HA appliances [24]. The SR is introduced to establish a communication technique for human to machine interaction, and has been the core challenge towards natural human to machine communication technology.

2.2 Historical Timeline of Speech Recognition

SR is a 20th Century invention.The historical timeline in the evolution of SR systems are summarized in Figure 2.2.

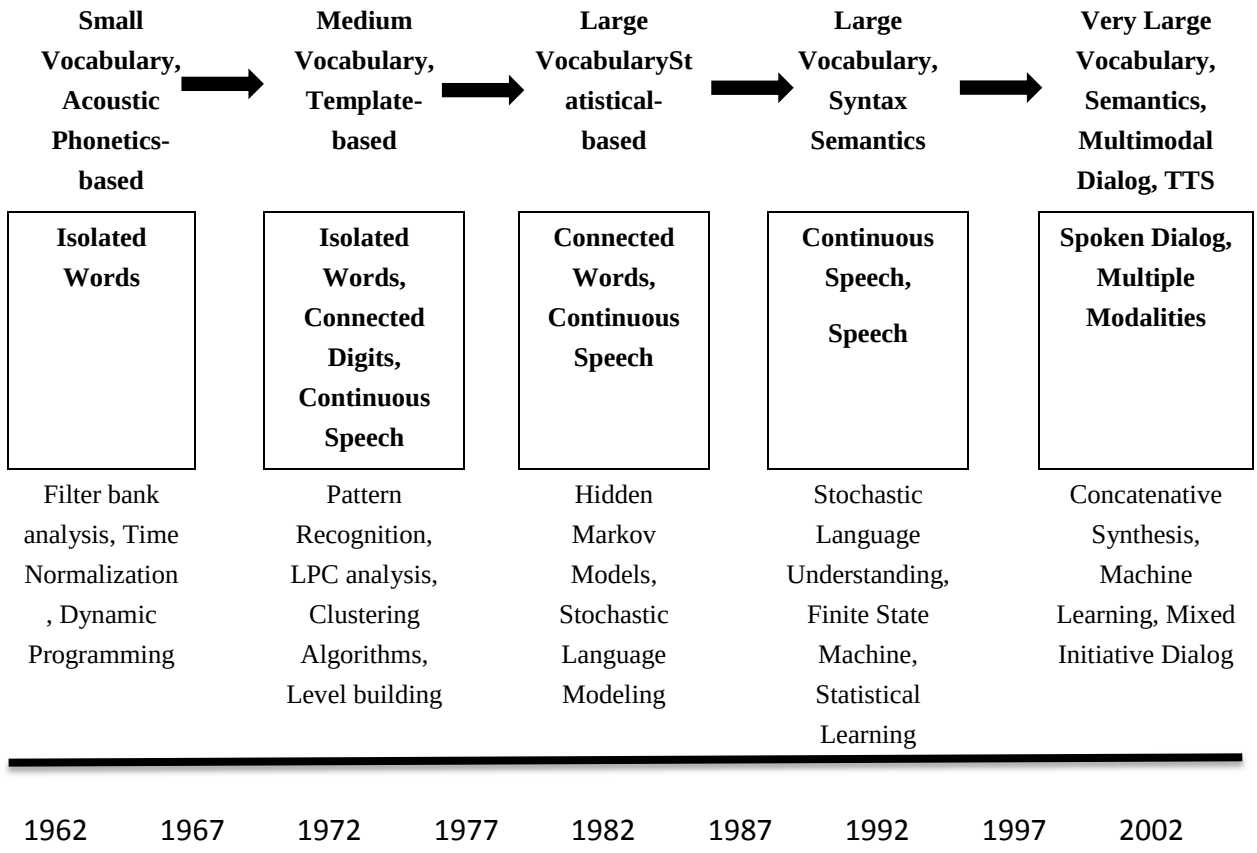

Figure 2.2: Milestones in SR and understanding technology over the past 40 years from [14, 25, 26]

From Figure 2.2, it can be seen that numerous techniques have evolved from 1962 to 2002 for 'isolated words', 'connected words', 'continuous words' and 'spoken dialog'.

2.3 Speech Recognition Parameters

The information provided by Deller et al. [14] about parameters associated with SR is summarized in the following sections to provide an understanding of the complexity and specifications of the SR issue.

2.3.1 Speakers

Speaker types can categorize as SD or SI. A SD system is designed to operate for a single or particular speaker. The speaker or group of speakers are trained and tested on the same speaker. On the other hand, a SI is designed to train many speakers to make the system capable of recognizing the speech of new speaker, and tested on speakers outside training population.

2.3.2 Utterances

SR can be classified into four (4) types of utterances to describe the type of speaking style to determine when the speaker starts and finishes an utterance.

1. Isolated Word

'Isolated word' recognizer usually requires a deliberate pause between each utterance. To simplify the end point detection, algorithm is needed to locate the beginning and the end of speech. This is fine for situations where the user is required to give only one word responses or commands. It accepts single words or single utterance at a time. This system work well and entirely appropriate in certain application such as 'command' and 'control' applications, in which the user requires the speaker to wait between utterances (usually doing processing during the pauses).

2. Connected Word

'Connected words' is similar to isolated words but allows separate utterance to "run-together" with a minimal pause between them.

3. Continuous Word

'Continuous word' allows users to utter in naturally spoken words. In continuous SR, the speech to be recognized is a sequence of words uttered in a fluent (continuous) manner. The confusion between different word sequences and more sophisticated recognizers to handle word boundaries issues grows as vocabulary grows large.

4. Spontaneous Word

This type of word is able to handle the variety of natural sounding and no rehearsal is required including mis-pronounciation, slight slutters and non words. Spontaneous speech contains disfluencies and more difficult to recognize.

2.3.3 Size of Vocabulary

Vocabularies (dictionaries) are lists of words that can be recognized by SR system [20]. Table 2.1 shows the size of vocabulary of SR.

Table 2.1: Size Of Vocabulary of SR systems [15, 26, 27]

Size	**Vocabulary**
Small	Ten of Words
Medium	Hundreds of Words
Large	Thousands of Words
Very Large	Ten of Thousands of Words

2.3.4 Recording

Sound is inherently an analog phenomenon. Microphone is used to capture sound waves. The resultant signal is converted to digital data using Analog to Digital Converter (ADC). Pulse Coded Modulation (PCM) is the format is used record the sound [27]. There are three (3) important factors need to be considered for recording process. The factors are Sampling Frequency, Sampling Resolution and Number of Channels.

i. Sampling Frequency, f_s

The sampling rate, sample rate or sampling frequency is defined as the number of samples per unit of time (usually seconds) taken from a continuous signal to a discrete signal. A sample refers to a value or set of values at a point and/or space. The standard sample rates and precision for audio samples can be viewed in detail in [19,

28, 29]. For this research, the optimal sampling frequency of 16 kHz has been used in all experiments.

ii. Sampling Resolution

Sampling resolution is important parameter in the digitization process of speech signal. It is used to represent each speech signal sample for storing each sample of speech as bit resolution. In this research work, all speech signals invariably use 16 bits/sample as bit resolution as stated in [30]. The number of bits/sample in turn depends on the number of quantization levels used during analog to digital conversion. The number of quantization levels will therefore be 2^{16}=65536 and are found to be optimal for preserving information present in the analog version of the speech signal.

iii. Number of Channels

Two types of channels can be used for recording the speech, i.e., mono and stereo. Table 2.2 elaborates the differences of both mono and stereo applicability, benefits and limitations.

Table 2.2: Mono versus Stereo

Mono Recording	Stereo Recording
• Stand for Monaural or monophonic sound	• Stand for stereophonic sound
• Done on one single channel	• Done on two separate channels composing of left and right sound input are independent of each other
• Cost is less for recording	• Cost is more expensive for recording
• Usage is widely on radio talk show, telephone hearing aid, mobile communication and Amplitude Modulation (AM) radio station	• Usage on movies, Television, Music Player and Frequency Modulation (FM) radio station
• Provides better quality of sound and excellent for reinforcing sound	• Does not provide quality of sound as mono but it provides a better experience when listening music, film and games
• All speech utterances recorded through singular audio channel so that audible cues only record the exact same signal as uttered.	• Provide more audible cues which are not only recording the speech utterances but also record the sound surrounding the area of recording.

Based on Table 2.2, it can be seen that Mono has a slight advantage compared to stereo in signal strength for the same power.

2.4 Related Work

Signal modeling represents the process to convert a sequences of speech signal sample into a set of vectors in probability space [31]. As stated earlier, there are three (3) signal modeling approaches to SR and they have different processes and analyses of FE and FM techniques. They are described in the following sub-sections.

2.4.1 Literature Review of SR for General Applications

Tables 2.3 to 2.5 provide the summary of SR Techniques for general applications using PR, AI and AP approaches respectively.

Table 2.3: Summary of SR Techniques using PR Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[32]	Mel Scale on LPC (MEL-LPC)	VQ HMM	Dictation system	SD type 20 speakers	'isolated English words' and 'connected English words'	Large	Wired microphone	97.2% and 96.6%
[33]	Average Magnitude Different Function (AMDF) LPCC	DTW	Design Speech Interactivity Embedded Module, SIEM	SD type 6 Speakers	English words	-	f_s = 8 kHz 16 Bit Condenser microphone	92.67% - 97.6%
[34]	Cepstral Analysis	DTW	Digit Recognizer	SD type 3 Speakers	5 English words Utter: 15 times/ speaker	Small	Wired Microphone	68%

Table 2.3 – Continued -

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[8]	MFCC	DTW HMM	Digit Recognizer	SD type 30 speakers	'isolated Malay words' (Kosong – sembilan) Utter: 10 times/ speaker	Small	f_s= 8 kHz Wired microphone	80.5%
[35]	LPCC MFCC LPCCCMS MFCCCMS	DTW VQ GA	Farsi word recognition	SD type 25 Speakers	Farsi words Clipped sentences from sports megazines	-	f_s =16 kHz 16 Bit Wired microphone	82.1% - 100%
[36]	MFCC	VQ DTW	Digit Recognizer	SD type 8 Speakers	Romanian words ('0'-'9') Utter: 3 times/ speaker	Small	Wired Microphone	96%

Table 2.4: Summary of SR Techniques using AI Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[9]	MFCC	MLP	Malay word recognizer	SI type 10 Speakers	'isolated Malay words' (25 words vocabulary) Utter: 10 times/ speaker	Small	f_s = 8 kHz Wired Microphone	84.73%
[37]	MFCC	ANFIS	Digit Recognizer	SI type 21 Speakers	'isolated Malay words' (Kosong – sembilan) Utter: 3 times/ speaker	Small	f_s =22 kHz 16 bit Wired Microphone	85.24%
[38]	Hybrid Wavelet 9/7 MFCC	BPNN	English word recognition	SD type 10 speakers	'isolated English words' (5 words) Utter: 10 times/ speaker	Small	Wired microphone	65% - 96%
[39]	LPC MFCC ZCR STE	ANN	Digit Recognizer	SD type 28 Speakers	'isolated English words' ('0' – '9')	Small	f_s= 22 kHz 16 Bit Wired microphone	37.5% - 85%

Table 2.5: Summary of SR Techniques using AP Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[10]	MFCC	HMM	Meal service Robot	3 Speakers	Chinese (25 words) Utter: 20 times/ speaker	Small	f_s= 16 kHz Wired Microphone	85% - 95%
[40]	Optimization of MFCC	HMM Toolkit	TIMIT Corpus	SI type 630 Speakers	English words (10 phonetic sentences)	Small	f_s= 16 kHz 16 bit Wired Microphone	78.66% - 90.68%
[41]	MFCC Bark Wavelet MFCC	HMM	English word recognition	-	30 single English words	Small	f_s= 11.025 kHz Wired microphone	93.7% - 96.3%

2.4.2 Literature Review of SR for Home Automation Applications

Tables 2.6 to 2.8 provide the summary of SR Techniques for HA applications using PR, AI and AP approaches respectively.

Table 2.6: Summary of SR Techniques using PR Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[12]	Beamforming	DDA	Smart HA system	SI type 21 Speakers	'continuous English words' (44 sentence) Utter 9 times/ speaker	Small	7 wireless microphone set near the ceiling	92.1%
[13]	LPCC	VQ	Manual Wheelchair Automator	SD type 21 Speakers	'isolated English words' (6 commands)	Small	f_s =8 kHz 16 bit Wired Condenser Microphone	75.8%

Table 2.6 – Continued –

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[42]	MFCC	V/NV Detection	Manual Wheelchair Automator	SD type 12 Speakers	Japanese words (12 commands)	Small	f_s= 16 kHz Microphone: Headset, Bone conduction, Pin and Bluetooth.	99.5%
[43]	MFCC	DTW	Hindi Key word recognition system	SD type 10 Speakers	hindi words (8 commands) Utter: 0 times/ speaker	Small	f_s = 16 kHz 16 bit Wired Ball Microphone	91.25% - 97.5%
[44]	-	DTW	Lift model control	SD type 1 speakers	8 Lithuanian words Utter: 100 times/ speaker	Small	f_s = 16 kHz Wired microphone	100%

Table 2.7: Summary of SR Techniques using AI Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[45]	LPC	ANN	Intelligent Wheelchair	SD type 2 Speakers	'isolated English words' (6 commands	Small	f_s = 11025 Hz 8 bit	90.3% - 93.3%

Table 2.8: Summary of SR Techniques using AP Approach

REF	FE	FM	APPLICATION	SR PARAMETERS				WA
				Speaker	**Utterance**	**Vocabulary**	**Recording**	
[46]	MFCC	LHEQ	English Recognizer	-	English words (50 digit strings and 209 english names)	Medium vocabulary	f_s= 8 kHz 16 bit wired and Motorola Bluetooth TM Headset	31.37% - 78.11%
[47]	MFCC	HMM Toolkit	English Recognizer	SI type 5 Speakers	'continous English words' (20 sentences)	Large vocabulary	Wired Microphone	85%

2.5 Critical Analyses

The following section provides critical analyses on SR for general applications based on Tables 2.3 to 2.5 and HA applications based on Tables 2.4 to 2.8.

For the SR techniques of general applications using PR approach, previous research work in [8] used ‘isolated Malay words’ with ‘word accuracy’ of 80.5% using MFCC and DTW, whilst researcher [32] work on ‘isolated Englih words’ has obtained ‘word accuracy’ of 97.2% using Mel-LPC and VQ . Other researchers [33, 34] have worked on English words where [33] has obtained ‘word accuracy’ 97.6% using LPCC and DTW and, [34] has obtained ‘word accuracy’ 68% using Cepstral Anlaysis and DTW. Researcher [35] have worked on Persian with ‘word accuracy’ of 100% using MFCCCMS and VQ-GA, and [36] have worked on Romanian with ‘word accuracy’ of 96% using MFCC and VQ-DTW. However, there is no information for ‘utterance’, one of the SR parameters for SR [33,34, 35, 36].

For the SR techniques of general application using AI approach, previous research work in [9] and [37] used ‘isolated Malay words’ with ‘word accuracy’ of 84.73% using MFCC and MLP and using MFCC and ANFIS has obtained ‘word accuracy’ of 85.24 respectively. Researcher [38] has worked on English words, with ‘word accuracy’ of 65% - 96% using Wavelet-9/7-MFCC and BPNN. Researcher [39] also has worked on English words with ‘word accuracy’ of 37.5% to 85% using LPC-MFCC-ZCR-STE and ANN. Both researchers [38] and [39] used ‘isolated English words’.

For SR techniques of general application using AP approach, previous research work in [40] was on English with ‘word accuracy’ of 78.66% - 90.68% using MFCC and HMM Toolkit. Researcher [41] worked on English with ‘word accuracy’ of 93.7% - 96.3% using Bark Wavelet-MFCC and HMM. Other researcher [10] has worked on Chinese with ‘word accuracy’ of 85% – 95%% using MFCC and HMM. However, there is no information for ‘utterance’, one of the SR parameter.

It was found that the SR techniques for general application using PR, AI and AP approaches are confined to wired speech input device only. In addition, the use of ‘connected Malay words’ is also limited. Amongst all the three approaches, it can be seen that PR approach yields better performance in terms of ‘word accuracy’ of 100%.

For SR techniques of HA applications using PR approach, previous research

work in [12] using Beam Forming and DDA obtained a ‘word accuracy’ of 92.1% for ‘continuos English words’. Researcher [13] using LPCC and VQ has obtained ‘word accuracy’ of 75.8% for ‘isolated English words’. Researcher [42] has worked on Japanese words and obtained ‘word accuracy’ of 99.5% using MFCC and Voice / No Voice detection. Researcher [43] has worked on Hindi with ‘word accuracy’ of 91.25% - 97.5% using MFCC and DTW. Researcher [46] has worked on Lithuanian with ‘word accuracy’ of 100% using DTW. However, there is no information for ‘utterance’ for all researcher under this category except [12] and [13].

For SR techniques of HA applications using AI approach, previous research work in [45] obtained ‘word accuracy’ of 90.3% using LPC and ANN for ‘isolated English words’. For SR techniques of HA applications using AP approach, previous research work in [46] has obtained ‘word accuracy’ of 31.37% to 78.11% using MFCC-LHEQ. However,there is no information for ‘utterance’. Researcher [47] worked on ‘continuous English words’ with ‘word accuracy’ of 49.22% using MFCC and HMM Toolkit.

A ‘connected Malay words’ was never used in PR, AI and AP approaches for HA applications. The use of wireless speech input device is limited to researcher [12], [42] and [46] only. In addition, [42] and [44] have used PR approach with higher ‘word accuracy’ compared to other approaches.

Literature showed that the DTW which is parametric clustering algorithm was popular due to its simplicity and scalability, whilst VQ was widely used in many applications with excellent unsupervised learning procedure. The performance of VQ technique depends on the existence of a good codebook of representative vectors. There are varieties of clustering method in VQ using distortion functions such as squared Euclidean distance; Mahalanobis distance, Itakura-Saito distance and relative entropy. The Linde-Buzo-Gray (LBG) algorithm has limitations where the quantized space is not optimized at each iteration, and the algorithm is very sensitive to initial conditions.

However, the errors due to segmentation or classification of smaller acoustically variable units, such as phonemes can be avoided because templates for entire words could be constructed. The efficiency of DTW and VQ depends upon the stored templates. However, it also has the disadvantage that pre-recorded templates are fixed, so variations in speech can only be modeled by using many templates per

word, which eventually becomes impractical [6]. The shortcoming of this approach is that, it does not work efficiently in the presence of distorted patterns [48].

For thirty years, ANN have been used for difficult problems in PR such as the pattern analysis of brain waves, and have been characterized by a low signal-to-noise ratio; in some cases it was not even known what was signal and what was noise [49].

Overall, the literature review focused on SR for general and HA applications using PR, AI and AP approaches, SR parameters for utterance and speech input device. In terms of utterance under the SR parameters, in general applications, only a few researchers have used 'isolated English words' and 'isolated Malay words' and none in 'connected English words' and 'connected Malay words'. Under the HA, theres isn't any use of 'Malay words' at all.

Amongst all the three approaches, PR produced the best 'word accuracy' for general and HA applications. For many languages, the 'word accuracy' was between 68% to 100%, whereas, it was about 80.5% for ' connected Malay words'. Therefore, there is a need to improve the 'word accuracy' of 'connected Malay words' using different algorithms to be on par with ' connected English words'. In terms of speech input device, it was found that for general and HA applications, most of the work was confined to wired microphone. Only a few have used wireless microphone for HA, and none for general applications.Wireless speech input device is important for those with limited movement such as the elderly and people with physical disabilities who are living by themselves in a HA environment. The limited use of wireless in HA based on literature makes it an important area for research in SR.

2.6 Summary

This chapter has presented a literature review of SR field. It began with the fundamentals of SR, followed by historical timeline and parameters of SR. Subsequently, related work of SR in general and home automation applications were discussed. Then, critical analysis of the literature review was presented. In the next chapter, principles of SR and methodology of the experimental work would be discussed.

CHAPTER 3

SPEECH MODELING AND METHODOLOGY

This chapter discusses the Speech Modeling of SR and methodology that was used to carry out the research. The major elements such the language model, Feature Extraction and Feature Matching are described under the Speech Modeling section. This is followed by discussion on data acquisition, feature extraction and matching algorithms under the methodology section. Then, discussion on the performance index to measure the performance of SR is provided, and finally summary is provided.

3.1 Speech Modeling

The modeling of language model, FE and FM are discussed in the following sections.

3.1.1 Language Model

The language model is important in SR processing. The language model consists of two (2) elements which are grammar and vocabulary. The grammar is a set of words and phrases to be recognized by SR. It also contains sets of predefined combination of words. All the active grammar words make up the vocabulary. The vocabulary is a list of words the SR will be compared against the word spoken by utterances. It is made up of all the words in all active grammar [50]. Only words in the vocabulary can be displayed. Each language has its own unique structures.

Malay, also known as Bahasa Melayu or Bahasa Malaysia refers to a group of language that belongs to the Malayo – Polynesian branch of the Austronesian family of language [51]. On the other hand, English is based on the reconstruction of Proto Germanic which has evolved into German, English, Dutch, Afrikaans, Yiddish and Scandinavian languages [52]. Therefore, the background clearly shows that both Malay and English languages are not connected and different with each other. Most

words of Malay language are very short and consists of a combination of small linguistic unit and possible structure with two sound Consonants (C) and Vowel (V) include V, CV, VCC, CV, VC, CVC, CVV, CCVC, CVCC, CCCV and CCCVC [25,37, 51]. The English language has 26 letters and 44 phonemes; whereas, there are 25 letters and 34 phonemes in the Malay language.

Malay morphology is an area that studies the structures, forms and categorizations of words. A morpheme is the term used in the morphology and it is a combination of phonemes into a meaningful unit. A Malay word can be comprised of one or more morphemes. The processes of word formation in Malay language are in the form of primary words, derivative words, compound words and reduplicative words. It is a language of derivative which allows the addition of affixes to form a new word [51]. In English language, the process involves the changes in the phonemes according to their groups. It can also be noted that affixation is more common in Malay than in English. Affixes for deriving nouns include /peN/, / per/, /an/, /ke-an/, /per-an/ and /peN-an/ and verb inflection include /meN/, /ter/, /beR/ and /kan/. The detail can be reviewed in published papers [50-52]. Table 3.1 shows the spelling, phoneme structure, syllable structure and number of syllable in each word spoken in the malay language.

Table 3.1: Word syllable structure

Digit Spelling	Phoneme Sequence	Syllables	No. of Syllable in each word
Buka	/Bu/ - /ka/	CV – CV	2
Tutup	/Tu/ - /tup/	CV – CVC	2
Kuatkan Suara	/Ku/-/at/-/kan/-/Sua/- /ra/	CV – VC – CVC – CVV- CV	5
Perlahankan Suara	/Per/-/la/-/han/-/kan/-/Sua/-/ra/	CVC – CV – CVC – CVC – CVV – CV	6
Tukar Siaran	/Tu/-/kar/ - /Sia/ -/ran/	CV – CVC – CVV – CVC	4

3.1.2 Feature Extraction

Feature Extraction consists of MFCC and CMS.

3.1.2.1 Mel Frequency Cepstral Coefficients (MFCC)

The objective of Feature Extraction is to extract the characteristic of the speech signal. The MFCC is based on the known variation of the human ear's critical bandwidths with frequency over 1 KHz, filters spaced linearly at low frequencies below 100 Hz and logarithmically at high frequencies above 1 KHz that have been used to capture the phonetically important characteristics of speech. MFCC consists of seven (7) computational steps as shown in Figure 3.1.

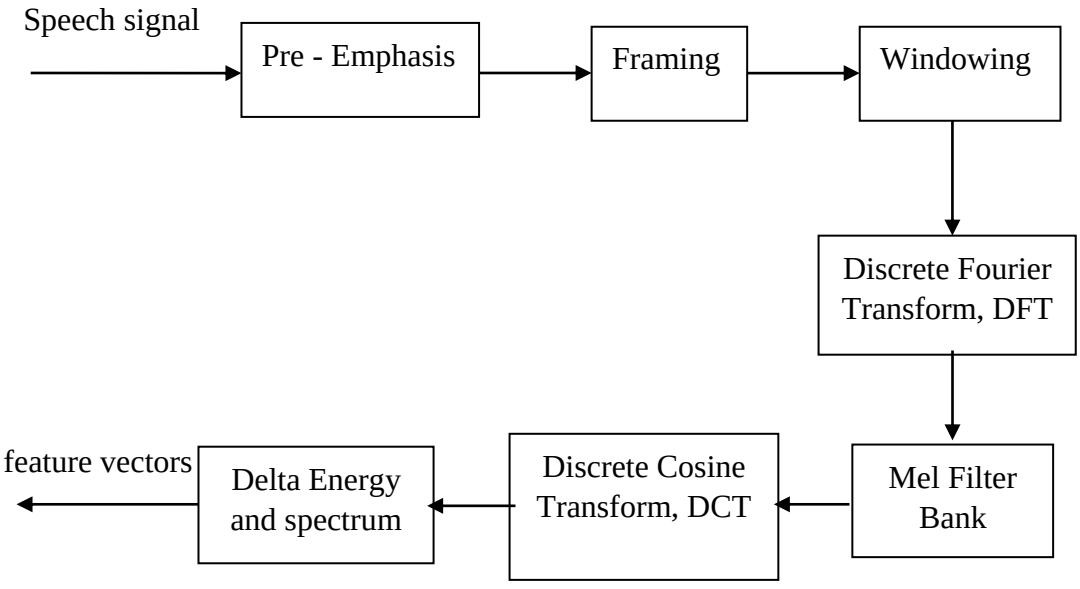

Figure 3.1: MFCC block Diagram [53 - 55]

1) Pre Emphasis

The process of signal passing through a filter which emphasizes higher frequencies. This process will increase the energy of signal at higher frequency using equation (3.1) [53].

$$Y(n) = X(n) - 0.95X(n-1) \quad (3.1)$$

where $X(n)$ = speech signal

$Y(n)$ = output signal after pre emphasis process

n = represents sequence number of samples

2) Framing

The process of segmenting the speech samples obtained from Analog to Digital Conversion (ADC) into a small frame with the length within the range of 20 to 40 msec. The speech signal is divided into frames of *N* samples. The Adjacent frames are separated by $M(M < N)$ where the sample rate per frame can be computed using equation (3.2) with overlapping region given in (3.3) [53, 54].

$N = 24ms$

$M = 12ms$

$$N, frame\ size = N * \frac{f_s}{1000} \quad (3.2)$$

$$M, Overlap\ Size = \frac{1}{2} * N * \frac{f_s}{1000} \quad (3.3)$$

3) Windowing

The signal needs to be changed into small parts in order to minimize the effect of spectral distortion that results from framing process. Windowing is used to minimize the signal discontinuities at the beginning, and the end of each frame. Then, the window is defined as $\omega(n)$ in equation (3.5) where:

$$0 < n < N - 1 \quad (3.4)$$

N = total number of samples in frame

$Y(n)$ = output signal

$Y'(n)$ = input signal after framing process

$\omega(n)$ = Hamming window

Then the result of windowing signal is shown below in equation (3.5) [53] :

$$Y(n) = Y'(n) * \omega(n) \quad (3.5)$$

Hamming window is used as window shape by considering the next block in feature extraction process. Hamming window is described in equation (3.6) [53].

$$\omega(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & 0 \ll n \ll N-1 \\ 0 & otherwise \end{cases} \quad (3.6)$$

4) Discrete Fourier Transform (DFT)

DFT is used to convert each frame of N samples from time domain into frequency domain. The Fourier Transform (FT) is to convert the convolution of the glottal pulse $U[n]$ and the vocal tract impulse response $H[n]$ in the time domain [53,56,57]. This statement supports the equation (3.7) and (3.8) [53].

$$Y(\omega) = FFT\,[H(t) * X(t)]$$
$$Y(\omega) = H(\omega) * X(\omega) \quad (3.7)$$

where $X(\omega)$, $H(\omega)$ and $Y(\omega)$ are the FT of $X(t)$, $H(t)$ and $Y(t)$ respectively. Let $X[k]$ = complex number representing magnitude and phase of that frequency in original signal. Then we get the equation (3.8) where $X[k]$ is the FT of $X[n]$, [53].

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{\frac{-j2\pi kn}{N}} \quad (3.8)$$

5) Mel Filter Bank Processing

The frequencies range in DFT spectrum is very wide and the speech signal does not follow the linear scale. Figure 3.2 shows the non linear transformation where it illustrates the equally spaced values on mel frequency scale correspond to non equally spaced frequencies. A set of triangular filters are used equation (3.8) to compute weighted sum of filter spectral components so that the output of process approximates to a Mel scale. Each filter's magnitude frequency response is triangular in shape and equal to unity at the centre frequency, and decrease linearly to zero at center frequency of two adjacent filters [58]. Then, each filter output is the sum of its filtered spectral components.

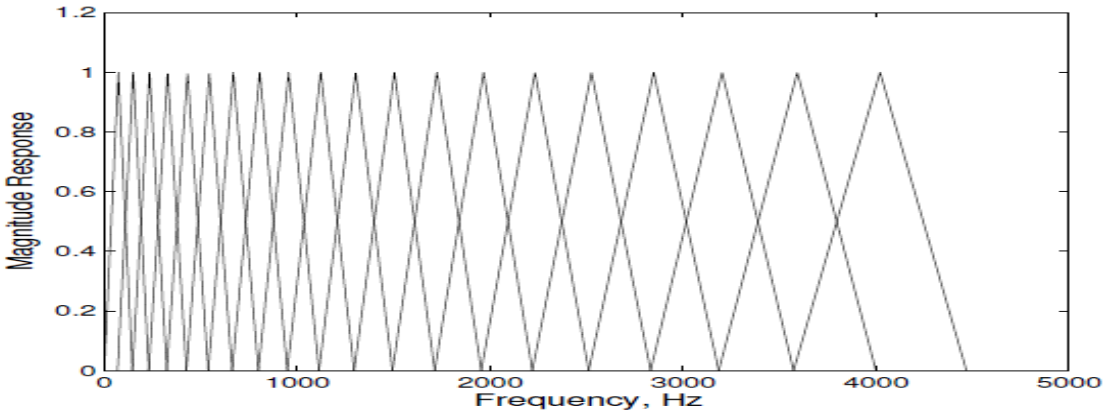

Figure 3.2: Mel scale filter bank [58]

The equation (3.9) is used to compute the Mel for given frequency f in Hz [53, 55].

$$Mel\ Scale = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \quad (3.9)$$

6) Logarithm and Discrete Cosine Transform (DCT)

The process to convert the log Mel spectrum into time domain using Discrete Cosine Transform (DCT) is called MFCC. This set of coefficient is called acoustic vectors. Therefore each input utterance is transformed into a sequence of acoustic vector. The MFCC is based on equation (3.10) [53] .

$$C[n] = \sum_{k=1}^{N-1} \log\left(\left|\sum_{n=0}^{N-1} X[n] e^{\frac{-j2\pi n}{N}}\right|\right) e^{\frac{j2\pi kn}{N}} \quad (3.10)$$

7) Delta Energy and Delta Spectrum

Since the speech signal and the frames change, features related to the change need to be added in cepstral features over time. Each of the 13 features (12 cepstral features plus energy), and 39 features of a double delta or acceleration feature need to be added. The energy in a frame for a signal X in a window from time sample, *t1* to time sample, *t2*, is represented at the equation (3.11) [53].

$$energy = \sum_{t=t1}^{t2} X^2[t] \quad (3.11)$$

Each of the 13 delta features represents the change between frames corresponding to the cepstral or energy feature, whilst, each of the 39 double delta features represents the change between frames in the corresponding delta features. Thus, the delta value $d(t)$ for a cepstral value $c(t)$ at time t can be estimated as equation (3.12) [53].

$$d(t) = \frac{C(t+1) - C(t-1)}{2} \tag{3.12}$$

3.1.2.2 Cepstral Means Subtraction (CMS)

A common problem involved in SR is transmission of channel that may vary from one recording session due to working environment of practical SR. CMS or Cepstral Mean Normalization (CMN) is common noise reduction technique addressing distortion. Once recognition occurs, the CMS will subtract the mean value from each feature vector, and then produce a normalized cepstrum vector which can capture the acoustic [54]. These technique is applied right after MFCC process to normalize the resultant features.

This methodology is applied in [55,56,58] successfully and also implemented for this research study. According to the reference [57], the ability of CMS to normalize the features is found to be very effective. This method reduced word error rate (WER) during matched and unmatched condition by performing filtering in the logarithmic spectral domain, and then, removed the mean of cepstral coefficient feature vectors over some interval. The process reduced the impact of stationary and slowly time varying distortion [25].

However, when computing the utterance mean,the CMS does not discriminate between voiced and unvoiced so that the mean is affected by the amount of noise included in the calculation. The other approaches that have been proposed to compensate the cross channel effects are Short Time Energy (STE) [13], Kernel filtering, ZCR [43] and Relative Spectra (RASTA) filtering [59] . CMS is used to compute the cepstral vectors and is useful for eliminating channel effect.

Firstly, considered a multidimensional signal $y(m,k)$, where m is the frame number of the feature, and each frame is with a duration of k samples of the speech signal. Assuming $y(m,k)$ can be decomposed into two mutually independent components, it can be represented as in equation (3.13) [60].

$$y\,(m,k) = x(m,k) + c(m,k) \tag{3.13}$$

where

$x(m,k)$ = clean speech observation sequence

$c(m,k)$ = noise speech observation sequence

Assuming that noise is stationary; therefore, $c(m,\ k)$ become constant which has no relation with m. Hence, equation (3.13) can be written as equation (3.14) [60].

$$y\,(m,k) = x(m,k) + c(k) \tag{3.14}$$

Then, consider the average of $x(m,k)$with respect to m frame, which is defined as equation (3.15) [60].

$$E\,[x(m,k)]_m \approx \frac{1}{N}\sum_{m=1}^{N} x(m,k) \tag{3.15}$$

where

N = Total frame number of speech signal

Each cepstral vector computes CMS feature vectors by subtracting the mean from each cepstral vector, then,

$$\begin{aligned} \acute{y}(m,k) &= y(m,k) - E\,[x(m,k)]_m \\ &= \{x(m,k) + c(k)\} - \{E\,[x(m,k)]_m + c(k)\} \\ &= x(m,k) - E\,[x(m,k)]_m \\ &= \acute{x}(m,k) \end{aligned} \tag{3.16}$$

where $\acute{y}(m,k)$ is the mean normalized feature of $y\,(m,k)$ and from equation (3.16) [60], the mean normalizd of noisy observation sequence is equal with clean speech and has no relation with the noise [60]. This indicates that all features can be normalized to establish robust SR. In SR , robustness refers to as ‘even tough the quality of the input speech is degraded or the speech acoustical, articulatory or

phonetic in the training and testing environments is differ, good recognition accuracy need to maintain' [57].

3.1.3 Feature Matching

Feature Matching consists of DTW and VQ.

3.1.3.1 Dynamic Time Warping (DTW)

DTW algorithm is based on Dynamic Programming techniques as described in [61]. These algorithm is used for measuring similarity between two time series which may vary in time or speed. DTW works based on finding the optimal alignment between two times series if one time series may be "warped" non-linearly by stretching or shrinking it along its time axis. In order to determine the similarity between the two time series or to find the corresponding regions between the two time series, the warping between two time series can used. Figure 3.3 shows the example of how one times series is 'warped' to another [62].

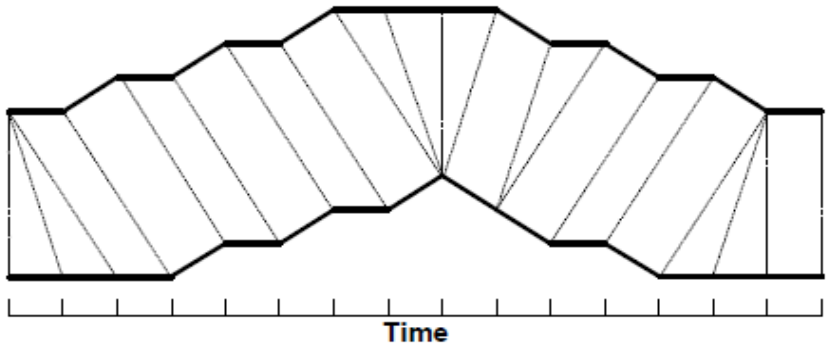

Figure 3.3: A warping between two time series [62]

As shown in Figure 3.3, each vertical line connects a point in one time series to its correspondingly similar point in the other time series. The vertical lines between them can be viewed more easily when the lines have similar values on the y-axis have been separated. Due to no warping would be necessary to 'line up' the two time series, all of the lines would be straight vertical lines if both of the time series were identical. After they have been warped together, the warp path distance measures the difference

between the two time series. The warp path distance is measured by the sum of the distances between each pair of points connected by the vertical lines. Thus, two time series that are identical except for localized stretching of the time axis will have DTW distances of zero.

The principle of DTW is to compare and measure similarity of two dynamic patterns by calculating a minimum distance between them.

Suppose there are two time series *Q* and *C*, of length *n* and *m* respectively,as in equation (3.17) and (3.18) [62]:

$$Q = speech\ input\ of\ test\ signal = q_i\ , q_2\ , q_3, \dots q_i, \dots q_n \quad (3.17)$$

$$C = speech\ input\ of\ template\ signal = c_i\ , c_2\ , c_3, \dots c_j, \dots c_m \quad (3.18)$$

1) To align two sequences using DTW, an *n*-by-*m* matrix where the (i_{th}, j_{th}) element of the matrix contains the global distance , $D_{ij} = d(q_i, c_j)$ between the two points, q_i and c_j is constructed. Then, to get local distance between two (2) features *q* and *c* of speech signal *Q* and *C*, the absolute distance between the values of two sequences is calculated using the Euclidean distance computation as equation (3.19) [62].

$$d\left(q_i, c_j\right) = \sqrt{\left(q_i - c_j\right)^2} \quad (3.19)$$

2) Each matrix element (i, j) corresponds to the alignment between the points q_i and c_j. Then, accumulated distance is measured by equation (3.20) [62]:

$$D(i,j) = min[D(i-1, j-1), D(i-1, j), D(i, j-1)] + d(i,j) \quad (3.20)$$

In recognition stages, as shown in Figure 3.4 where the horizontal axis represents the time of test input signal, and the vertical axis represents the time sequence of the reference template.

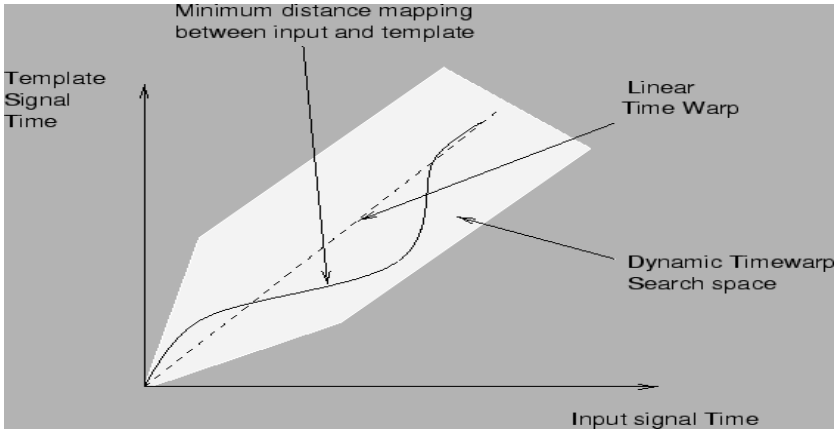

Figure 3.4: Example DTW [20]

The path shown is the result of the minimum distance between the input signal and template signal. The shaded area represents the search space for the input time to template time mapping function. The alternative assumption can be considered even with any monotonically non decreasing path within the space [18].

3.1.3.2 Vector Quantization (VQ)

VQ is a classical quantization technique. It is mapping of a large set of vectors into to a finite number of regions in that space. Each region is called a cluster or a codeword. Figure 3.5 shows the dots represent code vectors or centroid and each region or cluster is separated by the encoding regions. A set of all code vectors is called codebook whilst a set of encoding regions form the space partitions [63].

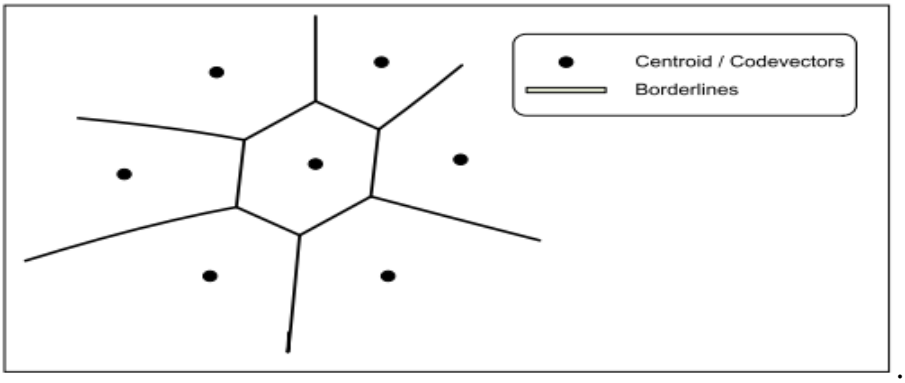

Figure 3.5: An example of codebook and codebook borderline of vector x

PR approach is basically divided into two parts, namely Pattern Training (PT) and Pattern Matching (PM). During recognition stage, once a minimum distance between dot and centroid have been produced, a set of codebook is then used to recognize an unknown speaker from trained spoken utterances. Then, to find the most nearest distance between code and the spoken utterances, Euclidean distance algorithm is used. The main concern of PT is the selection of feature vectors of the recorded speech samples, and training of the codebook randomly using the LBG VQ algorithm [63]. The VQ process involves the following steps :

1. Design a 1-vector codebook. This is the centroid of the entire set of training vectors and no iteration is required in this step. Set $m = 1$, and then, calculate centroid using equation (3.21) [63].

$$c_1 = \frac{1}{T} \sum_j^T X_j \tag{3.21}$$

2. Double the size of the codebook by splitting the codebook $Y(n)$, then divide each centroid C_i into two closed vectors as in equation (3.22) and (3.23) [63].

$$C_{2i-1} = C_i * (1 + \delta) \tag{3.22}$$
$$C_{2i} = C_i * (1 - \delta)$$

where $1 \leq i \leq m$ and δ = small fixed perturbation scalar. And Let $m = 2m$ then set $n = 0$ where n the iterative time.

3. Search Nearest Neighbour for each training vector

Find closest codeword in the current codebook. Assign that vector to the corresponding cell . Put X_j in the partitioned set P_i if C_i is the nearest neighbour of X_j

4. Find Average Distortion

After obtaining the partitioned sets, set $P = (P_i , 1 \leq i \leq m)$ and $n = n + 1$, calculate the average overall distortion using (3.23) [63].

$$D_n = \sum_{i=1}^{m} \sum_{j=1}^{T_i} \left(D_j^{(i)}, C_i \right) \tag{3.23}$$

where $P_i = \left\{ X_1^{(i)}, X_2^{(i)}, \ldots . X_{T_i}^i \right\}$

5. Update the Centroid

Each cell is update the codeword using the centroids of the training vectors assigned to that cell. Find centroids of all disjoint partitioned sets P_i by (3.24) [63].

$$C_i = \frac{1}{T_i}\sum_{j=1}^{T_i} X_j^{(i)} \tag{3.24}$$

6. iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold.

- If $\frac{D_{n-1}-D}{D_n} > \varepsilon$ where $\varepsilon = Threshold$

  Goto step 3 otherwise goto step 7.

7. Iteration 2: repeat steps 2, 3, and 4 until a codebook of size final codebook is reached

- If $m = N$, then take the codebook C_i as the final codebook otherwise go to step 2. Hence $N = codebook\ Size$

3.2 Methodology

The overall methodology for simulation of SR is given in Figure 3.6:

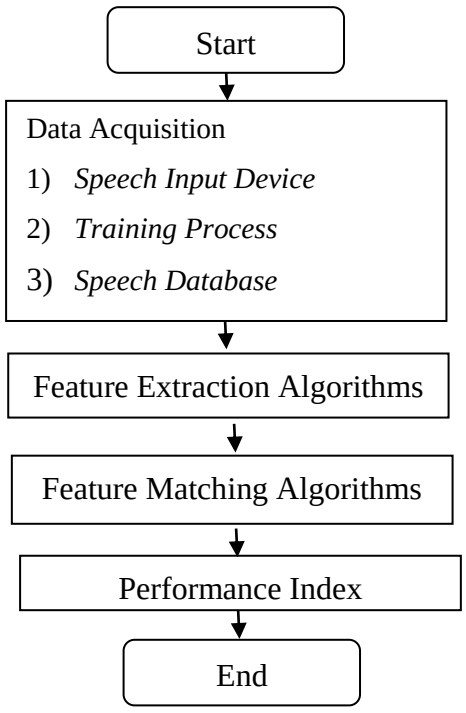

Figure 3.6: Overall Methodology

From Figure 3.6, it can be seen that SR involves Data Acqusition, FE, FM and Performance Measurement. A public domain software, GoldWave is used to acquire the speech signal. Then, the signal is converted into speech frames and analyzed by FE and FM algorithms. FE involves MFCC algorithms for producing feature coefficients, and then, normalized using CMS. DTW and VQ algorithms are applied in FM process at recognition stage. Finally, Performance Index is used for measuring the 'word accuracy'.

3.2.1 Data Acquisition

Data Acquisition involves input device, training process and speech dataset.

3.2.1.1 Speech Input Device

A microphone is used as the input device to capture the sound wave for SR. A microphone (colloquially called mic or mike) is an acoustic to electric transducer or sensor that converts sound into electrical signal. A wireless is general term about any type of microphone without a physical cable. Table 3.2 shows the specification of the wired and wireless microphones. In this work, a headset was used where the functionality is equivalent to a microphone.

Table 3.2 Headset Specification

WIRED MONO	WIRELESS STEREO
• Sonic Gear Hs 555 Headset	• Sony PS3 Wireless Stereo Headset Virtual surrounding 7.1
• Noise Cancellation Mic: High quality microphone features that is able to cancel the noise. • The detail is discussed in [64, 65]	• No noise Cancellation Mic
• Rechargeable Battery is not required	• Requires rechargeable battery
• Frequency response is 20 Hz to 20 kHz	• Frequency response is 2.4 GHz

The detailed specification of the headset can be found in Appendix A.

3.2.1.2 Training Process

Training process is a process to record and collect the speech data that would be used as training and testing dataset. The training of the data was called Template 1 (T1) and Template 2 (T2). For T1, the speaker needs to utter two times of each given five (5) words, and for T2, each word five (5) words needs to be uttered five times. This training was processed using the Gold Wave software. The details of both processes are summarized in Table 3.3.

Table 3.3 SR parameters for Training Process

T1	T2
1) Speaker : 15	1) Speaker : 15
2) Tools: 1.Wired Mono microphone 2. Wireless Microphone 3. Gold Wave software	2) Tools 1.Wired Mono microphone 2. Wireless Microphone 3. Gold Wave software
3) Environment: Laboratory with interrupted noise	3) Enviroment: Laboratory with interrupted noise
4) Utterance a) Repeat two (2) times each of the following English and Malay words respectively: 1. Switch On 1. Buka 2. Switch Off 2. Tutup 3. Volume Up 3. Kuatkan Suara 4. Volume Down 4. Perlahankan Suara 5. Change Channel 5. Tukar Siaran	4) Utterance a) Repeat five (5) times each of the following English and Malay words respectively: 1. Switch On 1. Buka 2. Switch Off 2. Tutup 3. Volume Up 3. Kuatkan Suara 4. Volume Down 4. Perlahankan Suara 5. Change Channel 5. Tukar Siaran
5) Sampling Frequency, Fs: 16 kHz with wave Pulse Coded Modulation (PCM) unsigned 16 bit	5) Sampling Frequency, Fs: 16 kHz with wave Pulse Coded Modulation (PCM) unsigned 16 bit

For T1, the training dataset is from the first utterance and the testing dataset is from the second utterance. For T2, the training dataset is from the first four utterances and the testing dataset is from the fifth utterance. The environment used for training process was a workplace with twenty (20) computers. Also there were a whiteboard, speakers and air conditioner. Interrupting noises that could be heard in this room are

working computers noise, air conditioner noise, convolutional noise from microphones and noise from speakers itself. Figures 3.7 to 3.8 show examples of training and testing datasets for the word “Switch Off”.

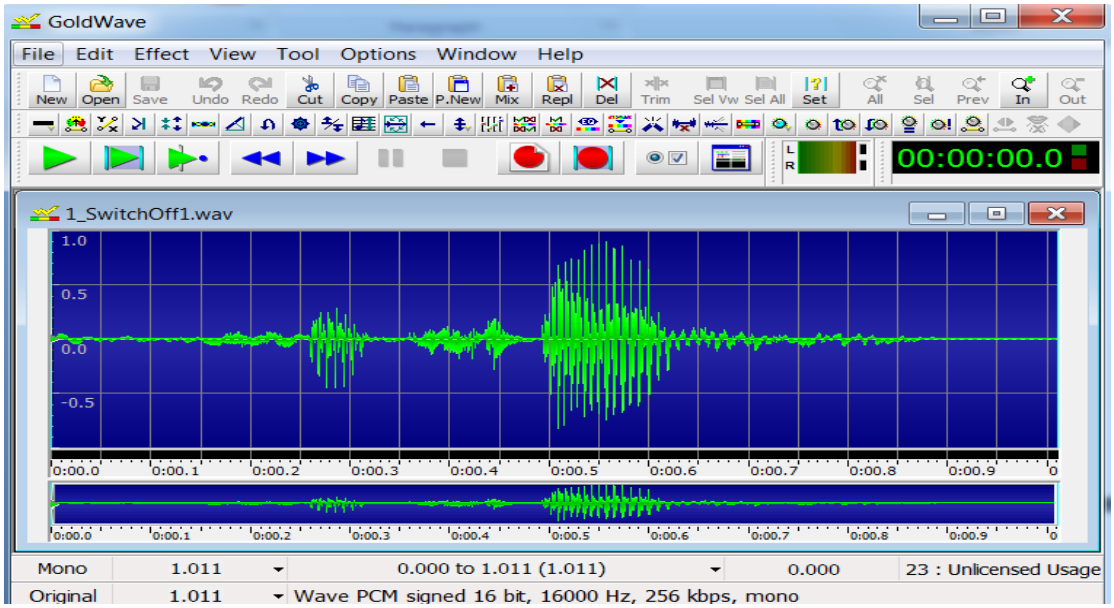

Figure 3.7: Training/Testing Dataset for Wired Headset

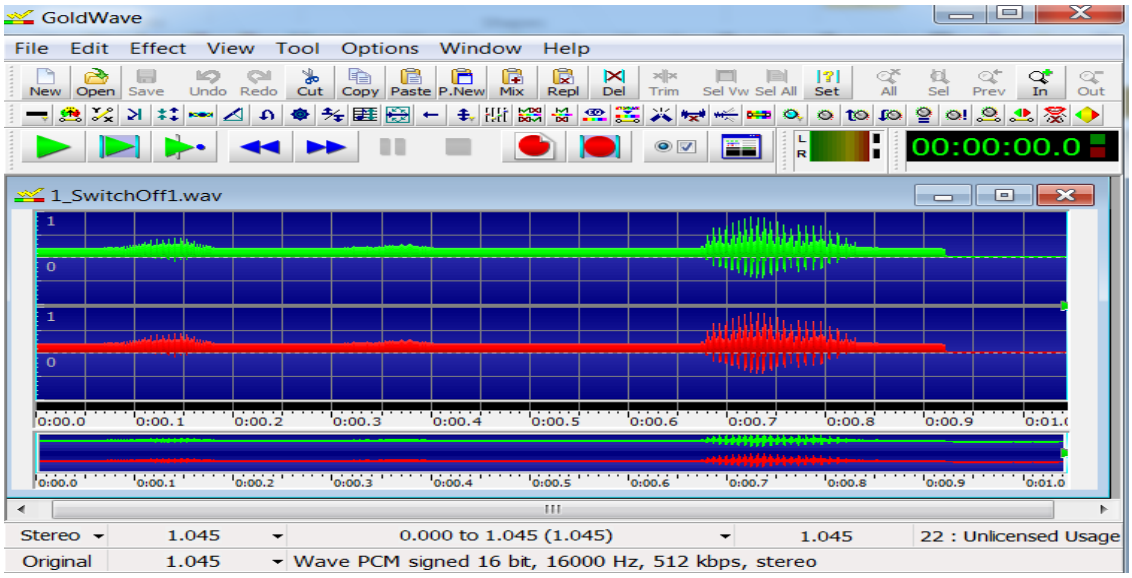

Figure 3.8: Training/Testing Dataset for Wireless Headset

As shown in Figure 3.8, the wireless headset produces two (2) amplitude sound wave signals of stereo channel type which represents the left and right channel respectively.

3.2.1.3 Speech Dataset

Upon completion of training process, the generated speech dataset file can be played back and saved as .wav. All data is then analyzed using MATLAB. The number of dataset per utterance for each training process is summarized in Tables 3.4 and 3.5 respectively. Tables 3.6 and 3.7 summarize the total recorded speech dataset for each utterance. The total number of samples for two (2) types of microphone used is summarized in Table 3.8.

Table 3.4: Number of Utterance Word for T1

Number of speakers	No. of Speech Samples Collected Per Utterance Word
15	2 times

Table 3.5: Number of Utterance Word for T2

Number of speakers	No. of Speech Samples Collected Per utterance Word
15	5 times

Table 3.6: Total Collected of Speech Samples for T1

Number of speakers	Total No. of Utterance Words * No. of Speech Samples Collected Per utterance Word * Numbers of speakers
15	5 * 2 times * 15
Total	150 Utterance Speech Samples

Table 3.7: Total Collected of Speech Samples for T2

Number of speakers	Total No. of Utterancet Words * No. of Collected Speech Samples Per Utterance Word
15	5 * 5 times * 15
Total	375 Utterance Speech Samples

Table 3.8: Total of Speech Samples using Microphones for T1 and T2

No. Of Training Process	No. Of Microphones * Total Utterance Speech Samples	Total Speech Samples
T1	2 * 150	300 samples
T2	2 * 375	750samples

Table 3.9 shows the total number of speech samples used for template and testing dataset . Since the first utterance of words is use as training dataset and the rest of data used for testing dataset.

Table 3.9: Total of Speech Sample for Training and Testing Dataset

No. Of Training Process	Training dataset for Wired Microphone	Testing dataset for Wired Microphone	Training dataset for Wireless Microphone	Testing dataset for Wireless Microphone
T1	75 samples	75 samples	75 samples	75 samples
T2	300 samples	75 samples	300 samples	75 samples

In this research, the language models used were ‘connected English words’ and ‘connected Malay words’. Table 3.10 shows the sequential numbers of the wave files assigned for the recorded speech samples for each utterance of ‘connected English words’.

Table 3.10: Wav files for Template Dataset for ‘connected English words’

Utterance Words	Wav files
Switch On	1_SwitchOn.wav through 15_SwitchOn.wav
Switch Off	1_SwitchOff.wav through 15_SwitchOff.wav
Volume Up	1_VolumeUp.wav through 15_VolumeUp.wav
Volume Down	1_VolumeDown.wav through 15_VolumeDown.wav
Change Channel	1_ChangeChannel.wav through 15_ChangeChannel.wav

Table 3.11 shows the sequential numbers of the wave files assigned for the recorded speech samples for each utterance ‘connected Malay words’.

Table 3.11: Wav files for Template Dataset for ‘connected Malay words’

Utterance Words	Wav files
Buka	1_Buka.wav through 15_Buka.wav
Tutup	1_Tutup.wav through 15_Tutup.wav
KuatkanSuara	1_KuatkanSuara.wav through 15_KuatkanSuara.wav
PerlahankanSuara	1_PerlahankanSuara.wav through 15_PerlahankanSuara.wav
Tukar Siaran	1_TukarSiaran.wav through 15_TukarSiaran.wav

3.2.2 Feature Extraction Algorithm

The FE Algorithm is described in Table 3.12.

Table 3.12: Feature Extraction Algorithm

Step	Description
Step 1:	Record the speech command using goldwave Software and save the file as wav format
Step 2:	Get MFCC feature vectors
Step 3:	Normalize the MFCC feature vectors to remove additional noise from transmission channel by using CMS and store into into the reference template. The output of this process then can be used to perform further analysis.

The overall process for **Step 2** is shown in Figures 3.9 to 3.15.

1) Pre Emphasis

The Matlab code for pre emphasis process is shown in the following:

Matlab Code : InputWave = filter ([1, -0.95], 1, InputWave);

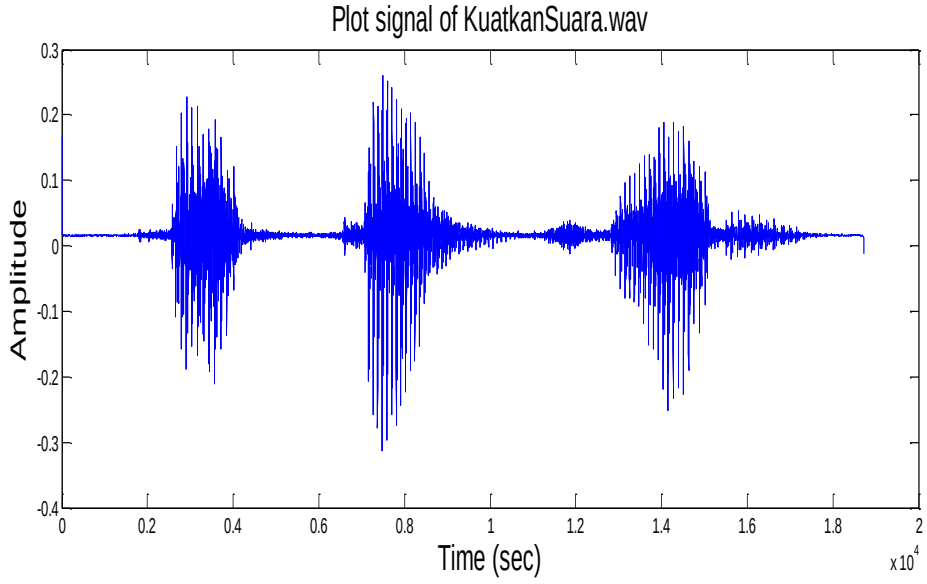

Figure 3.9: Waveform signal after Pre Emphasis

Figure 3.9 shows the speech waveform signal of test input word "*KuatkanSuara.wav*" which signifies the change in amplitude spectra over time. This step processes the passing signal through a filter to increase the energy of the signal at higher frequency.

2) Framing

The Matlab code for framing process is shown in the following:

Matlab Code: FrameSize_ms = 24;

Overlap_ms = (1/2)*FrameSize_ms;

FrameSize = round (FrameSize_ms*Fs/1000);

Overlap = round (Overlap_ms*Fs/1000);

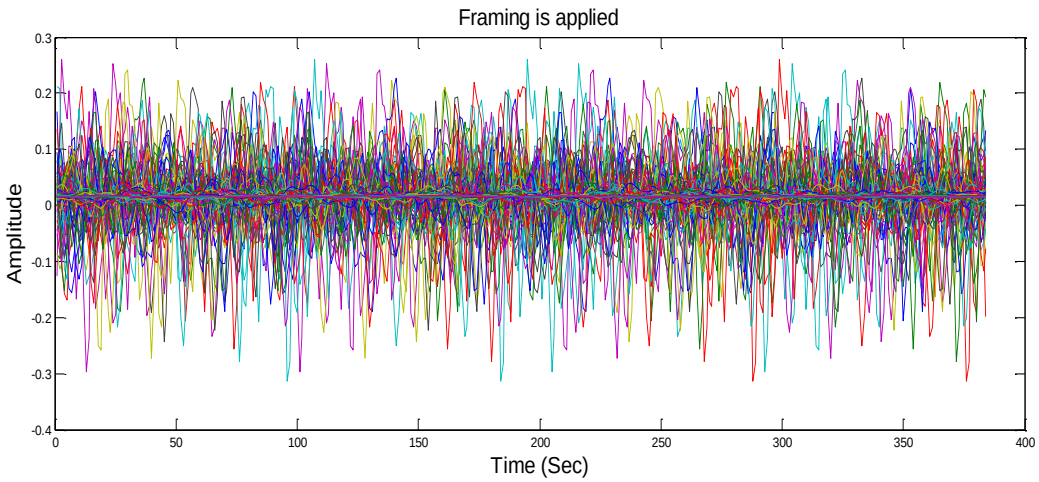

Figure 3.10: Waveform signal after Framing step

The speech signal is segmented into a small frame with the length of frame (frame size) 24 msec with overlap ½ of the frame size. The sampling frequency, *Fs* used is 16000 Hz and the frame size is (24 * 16000 / 1000) = 384 sample points, then, the frame duration is 384/16000 = 0.024 sec. Therefore, because the overlap is (12 * 16000)/ 1000 = 192 points then the frame rate is 16000/ (384 - 192) = 83.33 frames per sec. Figure 3.10 shows the speech waveform signal after framing step.

3) Windowing

The Matlab code for windowing process is shown in the following:

Matlab Code: WindowedFrame = hamming (FrameSize).*Frame (:,i);

Each frame has to be multiplied with window shape to keep the continuity of the first and the last point of the frame. In SR, the most commonly used window shape is the Hamming Window. HammingWindow can reduce the spectral leakage resulting from the convolution of the signal and decrease the possibility of higher frequency component in each frame [66, 67]. Figure 3.11 shows the speech signal after windowing step.

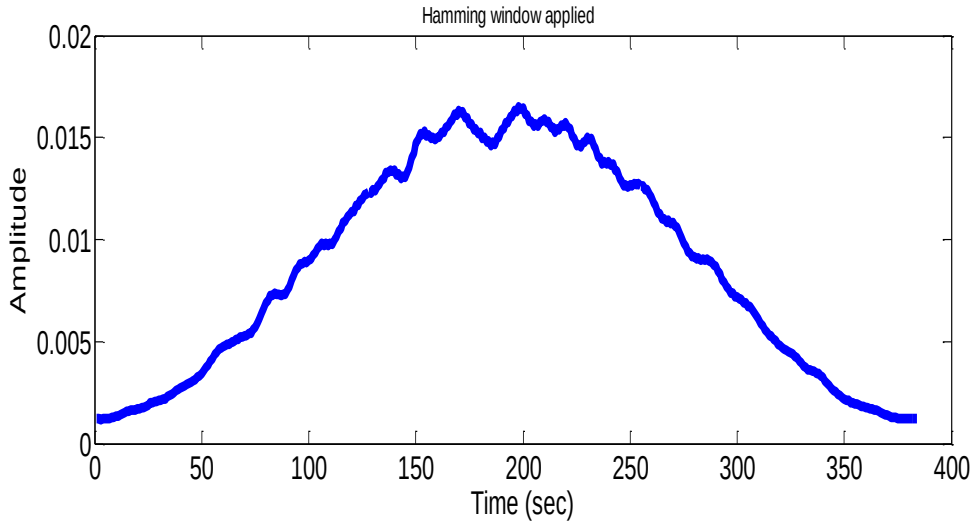

Figure 3.11: Waveform signal after Hamming Window is applied

4) Discrete Fourier Transform (DFT)

The Matlab code for DFT process is shown in the following:

Matlab Code: FFT_Frame = abs(fft(WindowedFrame));

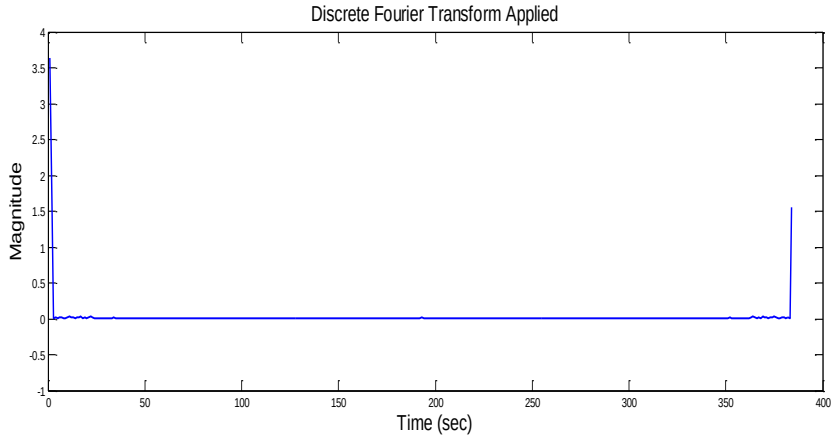

Figure 3.12: Waveform signal after DFT step

The next processing step is DFT, which computes the spectrum information of each window and obtains the magnitude of spectrum as shown in figure 3.12. Although DFT and Fast Fourier Transfrom (FFT) computation gives the same result but the difference is that DFT divides the signal into small frame, and then, takes transformation of the whole signal while FFT perform transformation on each frame separately [68, 69].

5) Mel Filter Bank Processing

The Matlab code for Mel Filter Bank process is shown in the following:

Matlab Code:

```
StartFreq=[1   3   5   7   9   11  13  15  17  19  23  27  31  35  40  46  55  61  70  81];
CenterFreq=[3  5   7   9   11  13  15  17  19  21  27  31  35  40  46  55  61  70  81  93];
StopFreq=[5    7   9   11  13  15  17  19  21  23  31  35  40  46  55  61  70  81  93 108];
No_of_FilterBanks = 20;
tbfCoef= TriBandFilter (FFT_Frame, No_of_FilterBanks, StartFreq, CenterFreq, StopFreq);
```

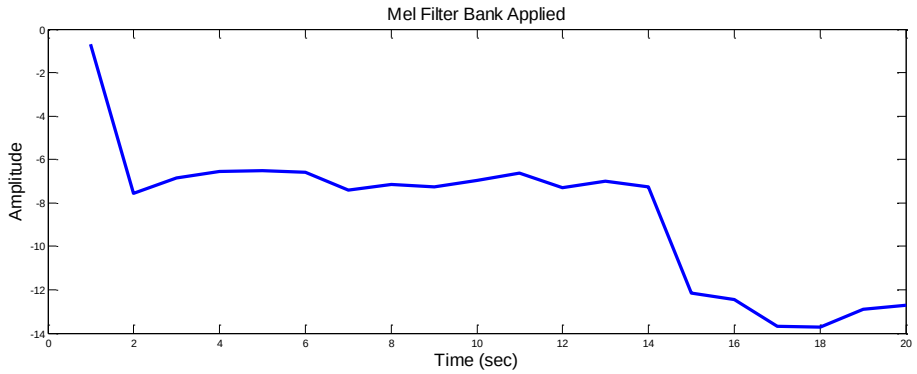

Figure 3.13: Waveform signal after Triangular Band Pass Filter step

The next step is the multiplication of the magnitude frequency response by a set of twenty (20) triangular band pass filters (Mel filter bank) as shown in figure 3.2 to get the log energy of each triangular band pass filter. The triangular band pass filter is used to reduce the size of features involved. The triangular of band pass filter parameter StartFreq, CenterFreq and StopFreq refer to the parameters used to build the 20 filter bank with proper spacing done by Mel scaling, whereas, each filter magnitude frequency response is triangular shape and equal to unity at the CenterFreq and decreases linearly to zero at CenterFreq of two (2) adjacent filters [70, 71]. Figure 3.13 shows the resultant signal after Mel filter bank processing is applied.

6) Logarithm and Discrete Cosine Transform (DCT)

The Matlab code for DCT process is shown in the following:

Matlab Code:

```
tbfCoef = log (tbfCoef. ^2);
No_of_Featurs = 12;
Cepstrums=Mel_Cepstrum2 (No_of_Featurs, No_of_FilterBanks,tbfCoef);
```

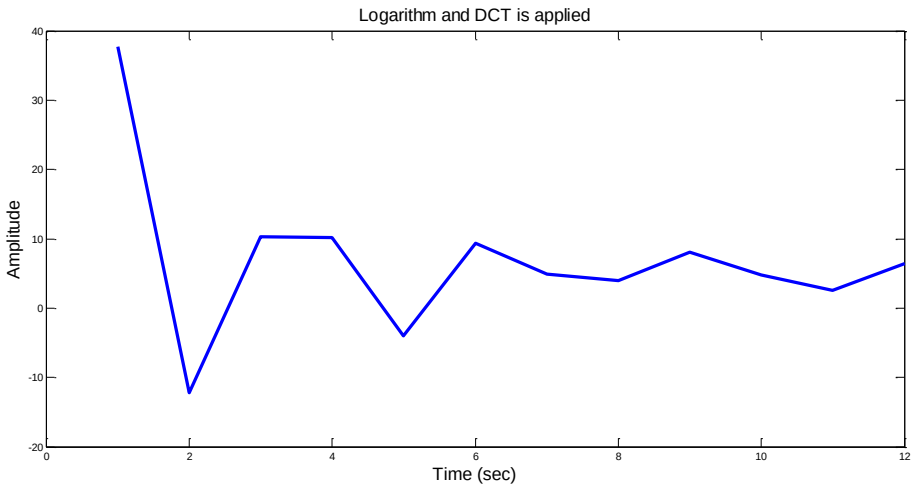

Figure 3.14: Waveform signal of 39 MFCC Coefficients

In this step, the log of base 10 is applied on each output cepstrums (spectrum) from Mel filter bank before applying DCT to make small values large enough, and large values small enough for DCT computation. Figure 3.14 shows the signal after DCT gives the final 12 MFCC feature vectors.

7) Delta Energy and Delta Spectrum

Below is matlab code for Delta energy and spectrum.

Matlab Code:

```
Featurs = [Featurs; logEnergy];
Delta_window = 2;
D_Featurs = DeltaFeature (Delta_window, Featurs);
```

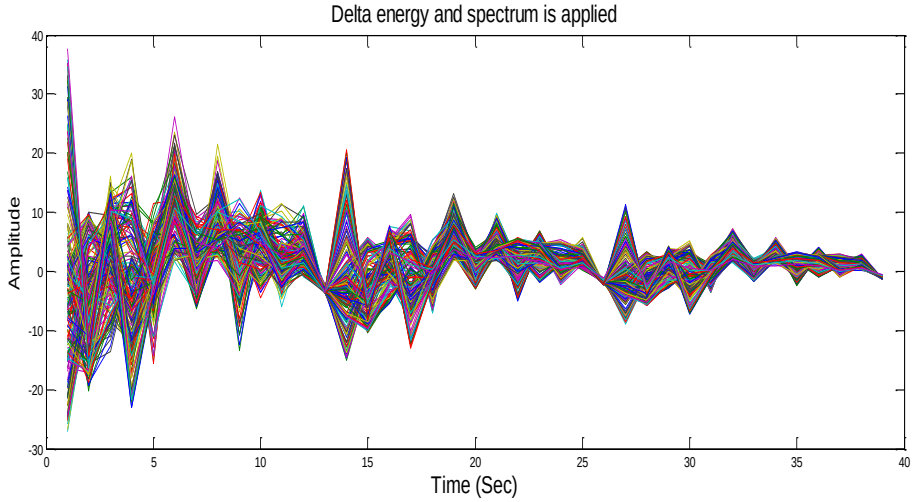

Figure 3.15: Waveform signal of 39 MFCC Double delta Coefficient

The performances of MFCC vectors can be increased by adding the log energy and by performing the delta operation within a frame, and usually done by adding the log energy as the 13^{th} feature to become 12 delta or velocity vectors (12 cepstral features plus energy). After adding both velocity and acceleration, 39 features are obtained as shown in figure 3.15.

3.2.3 Feature Matching Algorithm

The FM Algorithm is as shown in Table 3.13.

Table 3.13: Feature Extraction algorithm

Step 1:	Receive external speech command as Test input signal
Step 2:	Measure the similarity between training and testing input speech input based on DTW and VQ
Step 3:	If the Test input signal is matched with reference template, then, send the signal to activate the decision whilst If the Test input is not match with reference template then send the signal as Error.

The template generated in DTW algorithm is as the following:

1. A template is the representation of speech segment comprising a sequence of frame feature vectors, and each of the frames present information such as speaker characteristic, gender, dialect and etc.
2. First, all speech sample used for training data is extracted using MFCCCMS or MFCC, and then saved in template library as reference template as Mat file format. Each reference template is named differently to prevent redundancy and to be identified as unique speech sample. The reference template can be see in Appendix D.
3. Second, the test input is extracted and local match score distance is computed between the features frame by frame.Then, the lowest distance path through local distance matrix is found.
4. Then, the best match is found by selecting which frames of the reference template is the lowest distance path aligning the input pattern to the template such that the resulting error between them is minimized.

The VQ algorithm is as the following:

Step 1: Design a 1-vector codebook.

Step 2: Double the size of the codebook by splitting the codebook

Step 3: Search Nearest Neighbour for each training vector

Step 4: Find Average Distortion

Step 5: Update the Centroid

Step 6: Iteration 1: repeat steps 3 and 4 until distance falls below a preset threshold

Step 7: Iteration 2: repeat steps 2, 3, and 4 until a codebook of size final codebook is reached.

Once all the steps are performed for training, VQ codebook is generated.The Feature Matching algorihtms using DTW and VQ were implemented using MATLAB. The source codes for this can be viewed in Appendix B and C respectively.

3.3 Performance Index

The important role of SR is the ability to recognize speech test input even under influence of noise or substantial degradation factor. A set of training and testing dataset discussed in 3.2.1.3 are used for the evaluation of the performance of SR. The performance of SR can be measured using performance index. The most widely used measurement parameter are 'word error rate', (WER), and its complement, the accuracy or 'word accuracy', (WA).

The 'word error rate' is derived from the Levenshtein distance commonly used metric to evaluate SR. A 'word error rate' is defined as the sum of three (3) types of error divided by the total number of word in reference template. All three (3) errors are Substitution, Deletion and Insertion should be kept to a minimum to produce the good accuarcy [72]. Then 'word error rate' is computed as equation (3.25).

$$\text{WER} = \frac{S+D+I}{N} * 100 \tag{3.25}$$

where

S = number of Substitution (the reference word is replaced by another word)
Substitution error occur when the SR system incorrectly identifies a word from vocabulary (reference template).

D = number of Deletion (a word in the reference transcription is missed)
Deletion error occur when the SR system fail to respond to the spoken utterance even if the word is valid.

I = number of Insertion (a word is hypothesized that was not in the reference)
Insertion error may occur when the SR system respond to other source of sound other than spoken utterance as valid speech. The noise canceling microphone type can be use to reduce this error.

N = number of word in Reference template

The operator error occurs when the spoken utterance attempt to utter the other word that is not identifiable to the SR system. The minimum value of %WER rate is zero; means that there is no error. Thus, the automatic calculation of 'word accuracy' (WA), is a better representative for SR performance. A given set of recognition results requires the existence of reference transliterations for all spoken utterance. The percent 'word accuracy' is defined as equation (3.26).

$$\mathrm{WA} = \frac{N-S-D-I}{N} * 100 \tag{3.26}$$

This research discusses only accuracy and it is expressed as a percentage. The most important measurement of accuracy is whether the desired end spoken command is executed. Therefore, both 'word error rate' and 'word accuracy' are computed to note the relative error reduction to compare the performance of two algorithms. Both the performance indices are commonly used as SR evaluation performance [72-74]. The accuracy of the SR system is compared between HA applications and general applications using wired and wireless microphones.

3.4 Summary

This chapter has discussed the principles of SR and methodology that were used to carry out the research. The major elements such the language model for the ‘connected English words’ and ‘connected Malay words’, FE using MFCC and CMS, and FM using DTW and VQ were described under the speech modeling section. This was followed by discussion on data acquisition, algorithms for feature extraction and matching under the methodology section. Finally, the performance index to measure the performance of SR was provided. The findings and analysis of the experimental work are presented in Chapter 4.

CHAPTER 4

RESULT AND DISCUSSION

This chapter begins with the discussion on the experimental results using DTW. This is followed by description of experimental results using VQ. Subsequently, the analyses and discussion of all results are evaluated, and finally summary is provided.

4.1 Experimental Result for SR using DTW

A disadvantage of MFCC cepstral is that since it is a matrix, the length of the input and stored sequences would obviously be different if constant window spacing is used. Therefore, DTW is used to optimally align the variable length input and store the sequences of feature vectors based on 'some distance measure' between the features. Figure 4.1 shows the optimal warping path for *"KuatkanSuara.wav"*.

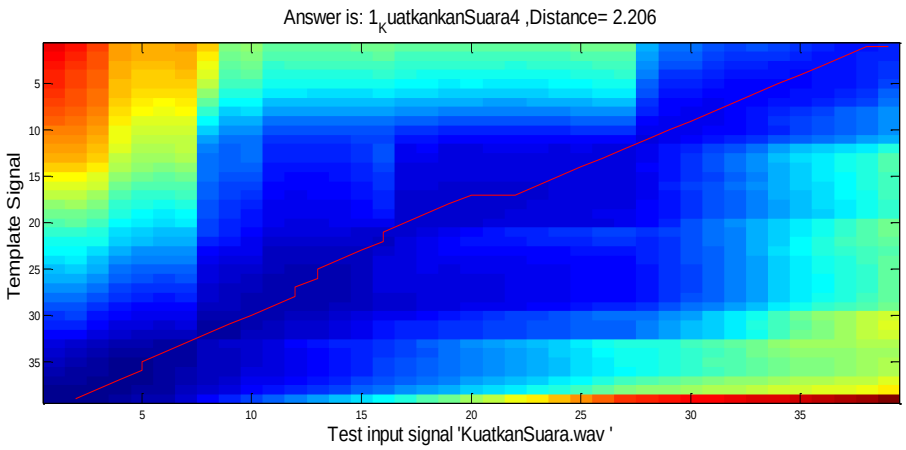

Figure 4.1: Optimal Warping Path for *"KuatkanSuara.wav"*

The result shown in Figure 4.1 confirms that the input test matched optimally with the reference template which was stored in the database as illustrated in Figure 3.4 of chapter 3. Once the best matching is found, the optimal path is achieved where word

"Kuatkan Suara" is executed. This finding is consistent with the principles of SR outlined in Chapter 3.1.3 where comparison of the template with the incoming input test was achieved via a pair wise comparison of the feature vectors in each. As discussed in [20], the total distance between the sequences is the sum or the mean of the individual distances between feature vectors. The purpose of DTW is to produce warping function that minimizes the total distance between the respective points of the signal. Furthermore, the accumulated distance matrix is used to develop mapping paths which travel through the cells with the smallest accumulated distances, and then the difference in the total distance between these two signals is minimized. Through this study, optimal warping path was achieved where the test input matched with the reference template as indicated by the path shown in figure 4.1.

4.1.1 Result for 'connected Malay words' using MFCCCMS and DTW techniques

This section provides the experimental results in recognizing 'connected Malay words'. In this experiment, the dataset consists of fifteen (15) speakers and utterance for each word. There are two (2) utterances for T1 and five (5) utterances for T2. The detail training and testing process are provided in section 3.2.1.2. The result was validated using the following example; if the user said "*Kuatkan Suara*", the engine returned '*Kuatkan Suara*, and if the "YES" action was executed, it is clear that the desired end result was achieved and be counted as one (1). However, if the engine returns other word, the result is not counted and it means it was not recognized. Table 4.1 shows the 'connected Malay words' using MFCCCMS + DTW.

Table 4.1: 'connected Malay words' using MFCCCMS + DTW

Word	Template 1, T1		Template 2, T2	
	Wired	Wireless	Wired	Wireless
"Buka"	14	12	14	14
"Tutup"	15	13	15	15
"Kuatkan Suara"	14	14	15	15
"Perlahankan Suara"	14	12	12	15
"Tukar Siaran"	15	13	15	15

Among all the words, word "*Buka*" and "*Perlahankan Suara*" produced lowest recognition. In T2, it can be seen that the word "*Perlahankan Suara*" produced 80% recognition for wired which is less than wireless where it achived 100% recognition. For the word "*Perlahankan Suara*" this has probably due to the fact that , malay is an agglutinative language and a new word can be created by adding affixes to a root word; for example, the word "*Perlahankan Suara"*. In this word, when adding two native prefix to one root base <per> and <an>, some spelling variation would have occured where the speaker utters with deletion of <r> be <pe> or <*pelahankan suara*> and deletion of <n> be <*perlahanka*> or <*perlahakan*> [75] . Then, 'word error rate', WER and 'word accuracy', WA are calculated to evaluate the result based on Table 4.1 by using equations (3.25) and (3.26) as discussed in section 3.3 of chapter 3. Tables 4.2 and 4.3 show results of accuracy using MFCCCMS+DTW techniques for T1 and T2 respectively.

Table 4.2: Accuracy using MFCCCMS + DTW for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 3$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+3+0}{75} * 100 = 4\%$	$Substitution, S = 0$ $Deletion, D = 11$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+11+0}{75} * 100 = 14.7\%$
WA	$\text{WA} = \frac{75-0-3-0}{75} * 100 = 96\%$	$\text{WA} = \frac{75-0-11-0}{75} * 100 = 85.3\%$

Table 4.3: Accuracy using MFCCCMS + DTW for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 4$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+4+0}{300} * 100 = 1.3\%$	$Substitution, S = 0$ $Deletion, D = 1$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+1+0}{300} * 100 = 0.3\%$
WA	$\text{WA} = \frac{300-0-4-0}{300} * 100 = 98.7\%$	$\text{WA} = \frac{300-0-1-0}{300} * 100 = 99.7\%$

From Table 4.2, it can be seen that the 'word accuracy' using wired is higher than using wireless microphone with 96%. Compared with Table 4.3, the wireless type is improved by %14.4 by increasing the number of reference template.

4.1.2 Result for 'connected Malay words' using MFCC and DTW techniques

Table 4.4 shows 'connected Malay words' using MFCC + DTW.

Table 4.4: 'connected Malay words' using MFCC + DTW

Word	**Template 1, T1**		**Template 2, T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
"Buka"	14	14	15	15
"Tutup"	15	15	15	15
"Kuatkan Suara"	15	14	15	14
"Perlahankan Suara"	15	13	13	14
"Tukar Siaran"	15	14	15	15

It can be seen from Table 4.4 that the word *"Perlahankan Suara"* and word *"Buka"* produced lowest recognition. In T2, it can be seen that the word "*Perlahankan Suara*" produced 86.7% recognition for wired which is less than wireless where it achieved recognition of 93.3%. Moreover, within a word, there will be variation in the length of

individual phonemes, *"Perlahankan Suara"* might be uttered with a long /e/ and short final /u/ or with a short /a/ and long /u/. Same goes word *"Buka"* might be uttered with a long /a/ and short final /u/ or with a short /a/ and long /u/.

Tables 4.5 and 4.6 show accuracy result using MFCC+DTW techniques.

Table 4.5: Accuracy using MFCC +DTW for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 1$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0 + 1 + 0 * 100}{75} = 1.3\%$	$Substitution, S = 0$ $Deletion, D = 5$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0 + 5 + 0}{75} * 100 = 6.7\%$
WA	$\text{WA} = \frac{75 - 0 - 1 - 0}{75} * 100 = 98.7\%$	$\text{WA} = \frac{75 - 0 - 5 - 0}{75} * 100 = 93.3\%$

Table 4.6: Accuracy using MFCC + DTW for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 2$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0 + 2 + 0}{300} * 100 = 0.7\%$	$Substitution, S = 0$ $Deletion, D = 2$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0 + 2 + 0}{300} * 100 = 0.7\%$
WA	$\text{WA} = \frac{300 - 0 - 2 - 0}{300} * 100 = 99.3\%$	$\text{WA} = \frac{300 - 0 - 2 - 0}{300} * 100 = 99.3\%$

Both tables 4.5 and 4.6 show that the results achieved were less than 10% error reduction in terms of WER for T1 with the wired input device, the 'word accuracy' is 98.7%, whereas, the wireless has 'word accuracy' 93.3%. Similar results were obtained for T2 where the wired and wireless obtained a 'word accuracy' of 99.3% respectively. These results conform to the requirement where at least 10% of WER reduction is recommended by researchers in [17,18] for a new algorithm to be considered for 'connected English words' using DTW.

4.1.3 Result for 'connected English words' using MFCCCMS and DTW techniques

This section provides the experimental results in recognizing 'connected English words'. Table 4.7 shows the 'connected English words' using MFCCCMS + DTW.

Table 4.7: 'connected English words' using MFCCCMS + DTW

Word	Template 1, T1		Template 2, T2	
	Wired	Wireless	Wired	Wireless
"Switch On"	15	13	15	14
"Switch Off"	14	15	15	15
"Volume Up"	15	11	15	15
"Volume Down"	15	13	15	15
"Change Channel"	15	15	15	14

English alphabet has fewer consonant letters than consonant sound, therefore a digraph like <ch>, <zh>, <th> and <sh> are used to extend the aplhabet letter [73]. Both word *"Switch On"* and *"Switch Off"* have digraphs. Usually, <ch> is pronouced <k> be < *Switk On* > or < *Swik Off* > . In addition, spoken utterances may have difficulities with final consonant digraphs either voiced <ch> or unvoiced <ch> in both words [77]. Both Tables 4.8 and Table 4.9 are show the results of 'connected English words' using MFCCCMS+DTW techniques.

Table 4.8: Accuracy using MFCCCMS a+ DTW for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 1$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+1+0}{75} * 100 = 1.3\%$	$Substitution, S = 0$ $Deletion, D = 8$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+8+0}{75} * 100 = 10.7\%$
WA	$\text{WA} = \frac{75-0-1-0}{75} * 100 = 98.7\%$	$WA = \frac{75-0-8-0}{75} * 100 = 89.3\%$

Table 4.9: Accuracy using MFCCCMS + DTW for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 0$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+0+0}{300} * 100 = 0\%$	$Substitution, S = 0$ $Deletion, D = 2$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+2+0}{300} * 100 = 0.7\%$
WA	$WA = \frac{300-0-0-0}{300} * 100 = 100\%$	$WA = \frac{300-0-2-0}{300} * 100 = 99.3\%$

From Table 4.8, the 'word accuracy' using wired is higher than using wireless microphone with 98.7%, whilst, in Table 4.9, wireless has increased the 'word accuracy' with 10% by increasing the number of reference template.

4.1.4 Result for 'connected English words' using MFCC and DTW techniques

Table 4.10 shows the 'connected English words' using MFCC and DTW.

Table 4.10: 'connected English words' using MFCC and DTW

Word	**Template 1, T1**		**Template 2, T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
"Switch On"	15	14	14	12
"Switch Off"	14	14	15	15
"Volume Up"	15	13	15	15
"Volume Down"	14	13	14	14
"Change Channel"	15	15	15	15

It is a known fact that within a word, the long vowel and short vowel pronunciation between speakers also varies. For example, as ashown in Table 4.10, the word *"Volume Up"* might be uttered with a long /O/ and short final /U/ or with a short /O/ and long /U/ [78]. This also same to words *"Volume Down"* and *"Switch On"*.

Tables 4.11 and 4.12 shows accuracy results using MFCC+DTW techniques.

Table 4.11: Accuracy using MFCC and DTW for T1

Accuracy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 2$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+2+0}{75} * 100 = 2.6\%$	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+6+0}{75} * 100 = 8\%$
WA	$\text{WA} = \frac{75-0-2-0}{75} * 100 = 97.3\%$	$\text{WA} = \frac{75-0-6-0}{75} * 100 = 92\%$

Table 4.12: Accuracy using MFCC and DTW for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 2$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+2+0}{300} * 100 = 0.7\%$	$Substitution, S = 0$ $Deletion, D = 4$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+4+0}{300} * 100 = 1.3\%$
WA	$\text{WA} = \frac{300-0-2-0}{300} * 100 = 99.3\%$	$\text{WA} = \frac{300-0-4-0}{300} * 100 = 98.7\%$

Table 4.1 and 4.4 showed that the recognition of word *"Perlahankan Suara"* for wired is less than wireless. This could be attributed to the ambient noise and lack of good speaking and recording environment. The length needed in order to keep the microphone element in the proper placement of the corner of mouth could be another factor. Therefore, it is significant to maintain distance between speaker and microphone at least 2 centimeters (cm) [79, 80]. In addition, it can seen from Tables 4.11 and 4.12 that the wired input device gave higher 'word accuracy' compared to wireless with 97.3% for T1 and 99.3% for T2. This could be attributed to the ambient noise and the wireless is not specifically designed for SR approach.

4.2 Experimental Result for SR using VQ

The VQ was chosen due to its simplicity, and the 'unsupervised learning procedure' make it easier to train and test the data with minimum computational time compared to DTW. The algorithm in LBG starts from one cluster to another, and separates into to two, four and so on until *N* clusters are generated; where, *N* is the desired number of clusters or codebook size (total number of training speech data). Additionally, VQ also encodes the speech patterns from the set of possible words into a smaller set of vectors to perform pattern matching. Therefore, selecting the more effective features instead of using the whole feature vectors can reduce the computational time. The command used to test the system is shown in Figure 4.2.

1) Matlab Code for training
   code = train (' D :\ train \ ', 8); where
   input1 = directory contains all train speech files
   input2 = number of train files in input1

2) Matlab Code for testing
   test ('D : \ test \ ' , 8 , code) where
   input1 = directory contains all test speech files
   input2 = number of test files in input1
   input3 = codebook of all trained speech data

Figure 4.2: Pseudocode for VQ algorithm

4.2.1 Result for 'connected Malay words' using MFCCCMS and VQ

This section provides the experimental results in recognizing 'connected Malay words'. In this experiment, the dataset consists of fifteen (15) speakers and utterance for each word. There are two (2) utterances for T1 and five (5) utterances for T2.

Table 4.13 shows the 'connected Malay words' using MFCCCMS + VQ.

Table 4.13: 'connected Malay words' using MFCCCMS + VQ

Word	Template 1, T1		Template 2, T2	
	Wired	Wireless	Wired	Wireless
"Buka"	13	14	13	15
"Tutup"	14	14	15	15
"Kuatkan Suara"	12	14	14	12
"Perlahankan Suara"	11	13	15	14
"Tukar Siaran"	11	14	14	14

Syllables in Malay is based on the consonent and vowels sequence as discussed on section 3.1.1. There are two rules in letter-to-sound (LTS) as listed in [79-80]; firstly, the deletion of the final <r> may be occur such in *"Tukar Siaran"* utter with <Tuka>. Secondly, the insertion of consonant <w> in vowel consonant <ua> appear twice in word *"Kuatkan Suara"* resulted <uwa>. In T1, it can be seen that the word "*Perlahankan Suara*" produced less recognition for wired with 73.3%, whereas wireless produced 86.7% recognition.

Both Tables 4.14 and 4.15 are show the results of 'connected Malay words' using MFCCCMS+VQ techniques.

Table 4.14: Accuracy using MFCCCMS + VQ for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 14$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+14+0}{75} * 100 = 18.7\%$	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+6+0}{75} * 100 = 8\%$
WA	$\text{WA} = \frac{75-0-14-0}{75} * 100 = 81.3\%$	$\text{WA} = \frac{75-0-6-0}{75} * 100 = 92\%$

Table 4.15: Accuracy using MFCCCMS+ VQ for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 4$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+4+0}{300} * 100 = 1.3\%$	$Substitution, S = 0$ $Deletion, D = 5$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+5+0}{300} * 100 = 1.7\%$
WA	$\text{WA} = \frac{300-0-4-0}{300} * 100 = 98.7\%$	$\text{WA} = \frac{300-0-5-0}{300} * 100 = 98.3\%$

From Table 4.14, it can be seen that the wired is lower than using wireless with 18.7% of WER. However, this can be improved by increasing the number of template T2.

4.2.2 Result for 'connected Malay words' using MFCC and VQ

Table 4.16 shows 'connected Malay words' using MFCC + VQ.

Table 4.16: 'connected Malay words' using MFCC + VQ

Word	**Template 1, T1**		**Template 2, T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
"Buka"	15	13	15	14
"Tutup"	15	13	15	15
"Kuatkan Suara"	14	14	15	13
"Perlahankan Suara"	15	13	15	12
"Tukar Siaran"	15	15	15	15

Each speakers have their own speaking style especially during the utterance of two or more syllable. In [81], stress shift refers to an independent variable to the change in the placement of stress or tone influenced by behavior of speaker utterance. The stress may be placed on the first syllable or occurs in final syllable. The word *"Perlahankan Suara"* pronounced with stress shift on <*Per*> or <Su> and same goes for *"Kuatkan Suara"* with stress on <*Ku*> or <Su>.

Table 4.17 and 4.18 shows accuracy result using MFCC+DTW techniques.

Table 4.17: Accuracy using MFCC+ VQ for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 1$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+1+0}{75} * 100 = 1.3\%$	$Substitution, S = 0$ $Deletion, D = 7$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+7+0}{75} * 100 = 9.3\%$
WA	$\text{WA} = \frac{75-0-1-0}{75} * 100 = 98.7\%$	$\text{WA} = \frac{75-0-7-0}{75} * 100 = 90.7\%$

Table 4.18: Accuracy using MFCC+ VQ for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 0$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+0+0}{300} * 100 = 0\%$	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+6+0}{300} * 100 = 2\%$
WA	$\text{WA} = \frac{300-0-0-0}{300} * 100 = 100\%$	$\text{WA} = \frac{300-0-6-0}{300} * 100 = 98\%$

Both Tables 4.17 and 4.18 show the result achieved was less than 10% of error reduction of WER where for T1, wired has 'word accuracy' of 98.7%, whilst wireless has 'word accuracy' of 90.7%. The difference of 'word accuracy' for T2 using wired and wireless is about 2%.

4.2.3 Result for 'connected English words' using MFCCCMS and VQ

This section discusses the experiment result using VQ for 'connected English words'.

Table 4.19: 'connected English words' using MFCCCMS +VQ

Word	Template 1, T1		Template 2, T2	
	Wired	Wireless	Wired	Wireless
"Switch On"	9	15	11	14
"Switch Off"	13	13	15	14
"Volume Up"	12	14	14	15
"Volume Down"	14	14	14	14
"Change Channel"	15	13	15	15

Stressed syllables in English are defined as style of utterance spoken with are longer, louder and higher in pitch [82]. Generally, stressed syllables tend to be stronger for a word containing prefixes whether on the first syllable or lightly stressed. The word, *"Switch On"* tend to be stressed <sswi> and *"Volume Up"* stressed <voo>.

Tables 4.20 and Table 4.21 are show the results of 'connected English words' using MFCCCMS+VQ techniques.

Table 4.20: Accuracy using MFCCCMS +VQ for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 12$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+12+0}{75} * 100 = 16\%$	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+6+0}{75} * 100 = 8\%$
WA	$\text{WA} = \frac{75-0-12-0}{75} * 100 = 84\%$	$\text{WA} = \frac{75-0-6-0}{75} * 100 = 92\%$

Table 4.21: Accuracy using MFCCCMS + VQ for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+6+0}{300} * 100 = 2\%$	$Substitution, S = 0$ $Deletion, D = 3$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+3+0}{300} * 100 = 1\%$
WA	$\text{WA} = \frac{300-0-6-0}{300} * 100 = 98\%$	$\text{WA} = \frac{300-0-3-0}{300} * 100 = 99\%$

It can be seen that results from Table 4.20 MFCCCMS +VQ achieved lower 'word accuracy' using wired for 'connected English words' for T1. However, in Table 4.21, the 'word accuracy' has increased with 14% by increasing the number of reference template. In addition, it can be noticed that the 'word accuracy' for wired is 1% lesser than wireless. Since the margin is too small, it can be deduced that both wired and wireless have similar values. The most important criteria have been met by obtaining 'word accuracy' of more than 90%.

4.2.4 Result for 'connected English words' using MFCC and VQ

Table 4.22 shows the 'connected English words' using MFCC + VQ.

Table 4.22: 'connected English words' using MFCC + VQ

Word	**Template 1, T1**		**Template 2, T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
"Switch On"	12	14	14	13
"Switch Off"	15	14	15	14
"Volume Up"	14	14	15	15
"Volume Down"	15	13	15	13
"Change Channel"	15	14	15	15

Syllables comprising a larger subword of vowel called nucleus is could be optionally prefixed and suffixed by one or more consonants [81]. Each utterance of the speakers uses different syllable boundries where it involves more than two syllables. For example, for the word, *"Switch On",* it could be spoken with boundries divison of <Swi--ch-On> or <Swit-chOn> which causes most of words not to be recognized [18].

Tables 4.23 and Table 4.24 are show the results of 'connected English words' using MFCC + VQ techniques.

Table 4.23: Accuracy using MFCC + VQ for T1

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 4$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+4+0}{75} * 100 = 5.3\%$	$Substitution, S = 0$ $Deletion, D = 6$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 75$ $\text{WER} = \frac{0+6+0}{75} * 100 = 8\%$
WA	$\text{WA} = \frac{75-0-4-0}{75} * 100 = 94.7\%$	$\text{WA} = \frac{75-0-6-0}{75} * 100 = 92\%$

Table 4.24: Accuracy using MFCC + VQ for T2

Accuarcy	Wired Microphone	Wireless Microphone
WER	$Substitution, S = 0$ $Deletion, D = 1$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+1+0}{300} * 100 = 0.3\%$	$Substitution, S = 0$ $Deletion, D = 5$ $Insertion, I = 0$ $Total\ word\ in\ template, N = 300$ $\text{WER} = \frac{0+5+0}{300} * 100 = 1.7\%$
WA	$\text{WA} = \frac{300-0-1-0}{300} * 100 = 99.7\%$	$\text{WA} = \frac{300-0-5-0}{300} * 100 = 98.3\%$

From Table 4.23, it can be seen that the 'word accuracy' ,%Wacc using wired is higher than using wireless microphone with 94%. Compared with Table 4.24, both

wired and the wireless type is improved by 5% and 6.3% of 'word accuracy' by increasing the number of reference template. It can be seen that results from Table 4.13, the word "*Perlahankan Suara*" produced lowest recognition for whilst Table 4.14 and 4.20 achieve lower 'word accuracy' using wired for 'connected English words' for T1. This could be due to recording process where the performance decreases significantly as soon as the microphone is moved away from the mouth of the speaker. This deterioration is due to a broad variety of effects including reverberation and presence of undetermined background noise.

4.3 Analyses and Discussion

Word pronunciation and perception are common tasks for human within the parameters of age group, differences in regional accents and the length of male and female vocal tracts. In daily communication, syllables in word are not pronounced clearly and sometimes does not have enough of acoustic information of the word. Humans, however can easily anticipate the incomplete information at perception level. In real time human communication, human does not listen to speech utterances in complete but anticipate them by comparing some existing sound model in their brain [85]. The main challenge of SR is to obtain an acceptable and higher accuracy for each algorithm in noisy enviroment.

Table 4.25 and Table 4.26 show the comparison of 'word accuracy' using both 'connected English words' and 'connected Malay words' respectively based on two (2) templates, T1 and T2.

Table 4.25: 'word accuracy' of 'connected Malay words'

Techniques	**'word accuracy' T1**		**'word accuracy' T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
MFCCCMS + DTW	96	85.3	98.7	99.7
MFCCCMS + VQ	81.3	92	98.7	98.3
MFCC + DTW	98.7	93.3	99.3	99.3
MFCC + VQ	98.7	90.7	100	98

From Table 4.25, the 'word accuracy' achieved by MFCC+VQ is higher for T1 and T2 using 'wired microphone' with 98.7% and 100% respectively. The MFCC+DTW produced higher 'word accuracy' using 'wireless microphone' for T1 with 93.3%, and

for T2 MFCCCMS+DTW produced ‘word accuracy’ with 99.7%. As seen from the table, since the initial results of T1 were not consistent, the number of words were increased in reference tempate of T2 for all techniques to improve the overall results.

Table 4.25 also shows that the ‘word accuracy’ for MFCCCMS+DTW has been improved from 85.3% to 99.7% for ‘wireless microphone’. The same goes for MFCCCMS+VQ where there has been an improvement from 81.3% to 98.7% for ‘wired microphone’. The results of the proposed techniques are consistent with the published work in [86] which used MFCC+CWRTs where CWRTs is the combination of ’Crosswords Reference Template and DTW’. This technique was used for SD with ten (10) ‘connected English words’ where the results has been improved from 85.3% (using traditional technique) to 99% for ‘wired microphone’.

Table 4.26 shows the ‘word accuracy’ of ‘connected English words’.

Table 4.26: ‘word accuracy’ of ‘connected English words’

Techniques	**‘word accuracy’ T1**		**‘word accuracy’ T2**	
	Wired	**Wireless**	**Wired**	**Wireless**
MFCCCMS + DTW	98.7	89.3	100	99.3
MFCCCMS + VQ	84	92	98	99
MFCC + DTW	97.3	92	99.3	98.7
MFCC + VQ	94.7	92	99.7	98.3

From Table 4.26, it can be seen that the ‘word accuracy’ for both MFCCCMS + VQ and MFCC + VQ techniques for ‘connected English words’ is more than 90% using ‘wired and wireless microphones’. In other words, error was reduced by 10%, and this is consistent with the findings by researchers in [17,18]. In addition, the theory that the ‘temporal and spectral variability’ is influenced by the size of codebook as advocated in [87] was successfully implemented and tested in this research where the ‘word accuracy’ was improved from 84% to 98% for MFCCCMS+VQ technique. Furthermore, MFCCCMS+DTW technique produced ‘word accuracy’ of 100%, which is higher than the previous mentioned study [32,36] with 96.6% and 96% respectively by using ‘wired microphone’. This shows that as more reference templates are used for the same word, higher the recognition rate.

The research findings of ‘connected Malay words’ and ‘connected English words’ shows that factors such as speaker, environment, acoustical and transmission system

influences the ‘word accuracy’. For example, speaker variation and the difference in transmission due to the use of ‘wired microphone’ could influence the outcome. This is evident from the ‘word accuracy’ of 84% that was achieved for T1. In addition, the use ‘wireless microphone’ which is of ‘stereo type’ is able to detect the surrounding; and this could increase the noise which possibly reduce the performance accuracy for MFCCCMS+DTW as indicated by the ‘word accuracy’ of 89.3%. Furthermore, it was found that it took longer computation time for DTW compared with VQ [62].

Table 4.27 shows a comparative analysis for HA application

Table 4.27: Overall ‘word accuracy’ for HA Applications

Ref	Techniques	Language	‘word accuracy’		Language	‘word accuracy’	
			Wired	Wireless		Wired	Wireless
Proposed Method	MFCCCMS + DTW	English	100	99.3	Malay	98.7	99.7
	MFCCCMS + VQ	English	98	99	Malay	98.7	98.3
	MFCC + DTW	English	99.3	98.7	Malay	99.3	99.3
	MFCC + VQ	English	99.7	98.3	Malay	100	98
[6]	Beamforming + Driven Decoding Algorithm,DDA	English	96.8%	-	Malay	-	-
[13]	LPCC + VQ	English	75.8%	-	-	-	-
[45]	LPC + ANN	English	90.3%	-	Malay	-	-
[46]	MFCC + Linear Histrogram Equalization LHEQ	English	79.61%	-	Malay	-	-
[47]	MFCC + HMM	English	85%	-	Malay	-	-

Table 4.27 shows that there is no previous work using ‘connected Malay words’ in HA applications. Therefore, the reseach that was carried out is useful in the development of the SR in HA application especially for ‘connected Malay words’.

The difference of 'word accuracy' between the 'connected Malay words' and 'connected English words' is only 1.3% by using wired. Similarly, for wireless, the difference is only 0.4% in terms of 'word accuracy'. Therefore, this indicates that the propsoed algorithms can be used in SR of 'connected Malay words' for HA.

Table 4.28 provides a comparative analysis of 'connected Malay words' in terms of the proposed techniques with other techniques.

Table 4.28: SR Performance for 'connected Malay words'

Ref	Techniques	Language	'word accuracy'	
			Wired	Wireless
	MFCCCMS + DTW	Malay	98.7	99.7
	MFCCCMS + VQ	Malay	98.7	98.3
	MFCC + DTW	Malay	99.3	99.3
	MFCC + VQ	Malay	100	98
[8]	MFCC + DTW & HMM	Malay	HMM = 90.7% DTW = 80.5%	-
[9]	MFCC + MLP	Malay	84.73	
[14]	LPC+MLP + LPC+DTW	Malay	78.09	-
[25]	MFCC + HMM	Malay	88.67	-
[37]	MFCC + ANFIS	Malay	85.24	
[88]	MFCC + ANN	Malay	90	-

The development of the SR in HA application especially for Malay language is important because study of SR for Malay is still infancy stage. The limitation of this research work is SD type compared with two previous work [9] and [37] where the 'word accuracy' was obtained with the different multi-layer neural network structures and Adaptive Neuro Fuzzy Inference System (ANFIS) respectively using SI approach and 'wired microphone' for 'isolated Malay words'. Both obtained 'word accuracy' of 84.73% and 85.24% respectively. This shows that the proposed technique achieved

performance relatively higher than both the previous works. In addition, similar previous work [8,14,25,88] based on SD type shows that the proposed method produced slightly higher performance.

The results from Table 4.28 proved the ability of common type 'wireless microphone' (not specifically designed for SR application) is able to produce better SR performance. Experimental work was carried out to observe the capabality of the 'wireless microphone' for SR in HA applications. This result was then compared with the performance of different brands of wireless microphone. This is shown in Table 4.29.

Table 4.29: Andre Audio Test Lab [89]

Wireless Devices	**Scores (%)**
Andrea BT – 200	94.1
Jawbone	84.2
Blue Ant Z9i	90.1
Plantronics Calisto	91.3
Plantronics Voyager 855	86.9
Logitech Cordless	83.9
Motorola H500	69.1
Sony PS3 Wireless Stereo Headset Virtual surrounding 7.1	**98% - 99.7%**

From Table 4.29, it can be deduced that the wireless microphone that was used in the experiment is within the acceptable specifications as speech input device in terms of frequency response which are in the range of 900MHz to 2.4GHz [90, 91].

Table 4.30 shows the ‘word accuracy’ of ‘connected English words’ in General applications.

Table 4.30: ‘word accuracy’ ‘connected English words’ in General Application

Ref	Techniques	Language	‘word accuracy’	
			Wired	Wireless
	MFCCCMS+ DTW	English	100	99.3
	MFCCCMS +VQ	English	98	99
	MFCC + DTW	English	99.3	98.7
	MFCC + VQ	English	99.7	98.3
[13]	LPCC + VQ	English	75.8%	-
[30]	MEL –LV+VQ	English	97.2	-
[31]	LPCC+DTW	English & Mandarin	80.5%	-
[32]	Cepstral analysis + DTW	English	68.33	-
[85]	MFCC+DTW	English	99	-
[89]	MFCC + DTW	English	90	-

Most researchers applied speech enhancement of data signal before FE to offset such a problem. The channel influence on the speech can be represented in the cepstral domain through an additive component to the cepstrum of the clean speech. In this case, to compensate the channel effect, the channel cepstrum can be removed by subtraction of the cepstral mean [92]. This theory can be proven as shown in Table 4.30 where the proposed method using MFCCMS achieved slightly better ‘word accuracy’ performance as compared with other previous work [13,30-32, 83, 89]. Since there is no previous work on wireless microphone, it can be inferred that the proposed method could be adopted as a new algorithm for SR application.

4.4 Summary

This chapter has discussed the results of SR by using the ‘word accuracy.’ The first section of this chapter presented the results of template method approach . The second section discussed the findings of proposed algorithms and the validation of the results for General and HA applications. In the next chapter, the research undertaken is summarized with discussions on contributions made and future work is outlined.

CHAPTER 5

CONCLUSIONS

In this thesis, a comprehensive discussion on MFCC, CMS, DTW and VQ for ‘connected Malay words’ and ‘connected English words’ using wired and wireless speech input device are provided. In section 5.1, a critical analysis of the main achievements of the research reported in this thesis is evaluated by discussing the research results and contributions with respect to the stated objectives. This is followed by some suggestions for future work in section 5.2, and finally it is concluded with remarks on future outlook of SR in section 5.3.

5.1 Critical Evaluation of Achievements

This research has adressed two critical issues for SR with ‘small vocabulary’ and ‘connected word’. First, it dealt with development of SR algorithms based on different approaches for ‘connected Malay words’ and ‘connected English words’. Secondly, it evaluated the performance of wired and wireless as speech input device.

To contextualize the research, analysis on literature review of the fundamental of SR and algorithms for improvement of SR in a noisy environment was presented in Chapter 2 where several aspects of SR approaches , history of SR, the parameters involved for recording of SR were discussed. Under the parameters, elements such speakers, utterance, size of vocabulary and recording were described. Subsequently, related works of SR in general and HA applications were provided. Then, critical analysis of literature review was presented.

The implementation of both the issues was addressed in Chapters 3 and 4 where principles of SR and the methodology for data acquisition that comprised of T1 and T2 were presented using wired and wireless speech input device. Then, Feature Extraction and Matching algorithms were applied. This is followed by

discussion on the experimental work using the proposed algorithms. Finally, it discussed the ‘word accuracy’ based on performance indices.

In this research, a different combination of algorithms of Feature Extraction and Matching such as MFCC, CMS, DTW and VQ have been used effectively. An ‘effective algorithm’ can be refered in [17,18] where it was stated that in order to adopt a new algorithm, a requirement of at least 10% of ‘word error rate’ or ‘word accuracy’ reduction is recommended. Research findings show that the ‘word accuracy’ for ‘connected Malay words’ using wireless input device is as the following; MFCCCMS+DTW produced 99.7%, MFCC+DTW produced 99.3%, MFCCCMS+VQ produced 98.3% and MFCC+VQ produced 98%. Similar results were obtained for ‘connected English words’ using the proposed algorithms.

Overall, the results show that the SR accuracy for algorithms of both languages was over 90%, and the error was reduced by at least less than 10%. Therefore, proposed algorithms can be used efficiently in the development of the SR in HA application especially for ‘connected Malay words’ because the study of SR for ‘connected Malay words’ is still at its infancy stage.

A comparative analysis shown in Table 4.29 demonstrates the ability of commercial wireless headset to provide good performance of SR. In addition, the results show that the proposed algorithms for ‘connected Malay words’ and ‘connected English words’ in general applications produce better ‘word accuracy’ compared with other researchers [8-9,13-14,25,30-32,37,83,88-89] for wired microphone. Since there is no previous work on wireless microphone, it can be inferred that the proposed algorithms could be used effectively for SR. The differences of ‘word accuracy’ for ‘connected Malay words’ and ‘connected English words’ was only 1.3% by using wired input device. Similarly, using wireless input device, the difference of ‘word accuracy’ is 0.4%. Therefore, it can be deduced that the proposed algorithms for ‘connected Malay words’ is comparable with ‘connected English words’ due to the close ‘word accuracy’.

Thus, the contributions of this research for SR in a noisy environment are summarised as the following:

1. Development of an effective FE and FM algorithms using wired and wireless microphones for ‘connected Malay words’ and ‘connected English words’.
2. The result show that wireless microphone is capable to be used as speech input device for SR in HA applications.

5.2 Suggestions for Future Work

A few suggestions for further research are given in the following(tambah syllablee):

I. Since this research concentrated on ‘small vocabulary connected word’, the usage is limited. This can be extended to ‘medium’ and ‘large’ vocabulary.

II. PR approach is widely used in solving many SR problems. Other approaches such as the AI and AP could be investigated further .

III. To explore the thorough effect of linguistic categories including the pronunciation variation in different languages syllables, phoneme inventory, lexical word and grammar.

IV. Since this work has produced promising results in terms of experimental work, these results could be applied practically by testing them out in a SR prototype. Therefore, SR circuit should be developed for this purpose using an integrated circuit (microcontroller or speech chip), memory, ADC-DAC, a PWM, general purpose I/O ports, amplifier, filter and other peripheral circuit and associated components.

5.3 Concluding Remarks

The findings emanate from this work has contributed to an improved understanding of the SR algorithms approach to develop an effective SR system for ‘connected English words’ and ‘connected Malay words’ using wireless microphone. In additon, this research also shows that there are different methodologies remains to be explored to produce a better system.

REFERENCES

[1] B.Gold and N.Morgan, *Speech and Audio Signal Processing*. John Wiley & Son Inc., 2000.

[2] M. Butt, M. Khanam and A. Khan et.al, "Controlling Home Appliances Remotely Through Voice Command," *International Journal of Advanced Computer Science and Application (IJACSA)*, Vol. 48 , No. 17 , pp. 35-39, July 21, 2012.

[3] L. Haddon, "Home Automation: Research Issue", in *2nd EMTEL Workshop*, pp. 1-20, November 10-11, 1995.

[4] Y. Krishna, S. Nagendram, "Zigbee Based Voice Control System for Smart Home," *International Journal Computer Technology and Application*, Vol. 3, No. 1, pp. 163-168, January 2012.

[5] V. Ramya and B. Palaniappan, "Embedded Home Automation for Visual Impaired," *International Journal of Computer Application.*, Vol. 41, No. 18, pp. 32-39, March 2012.

[6] B. Lecouteux, M. Vacher, F. Portet, "Distant speech recognition in a smart home: Comparison of several multisource asr in realistic conditions," in *12th Annual Conference of International Speech Communication Association INTERSPEECH*, Augast 27-31, 2011.

[7] M. Vacher, D. Istrate, T. Joubert,T. Chevalier, S. Smidtas, B. Meillon, B. Lecouteux, "The SWEET-HOME Project : Audio Technology in Smart Homes to improve Well-being and Reliance," in *33rd Annual International Conferences of IEEE EMBS*, pp. 5291-5294, August 30 - September 3, 2011.

[8] S. Al-Haddad and S. Samad, "Decision Fusion for Isolated Malay Digit Recognition Using Dynamic Time Warping (DTW) and Hidden Markov Model

(HMM),” in *The 5th Student Conference on Research and Development – SCOReD*, December 11-12, 2007.

[9] N. Seman, Z. Bakar and N. Bakar, “Measuring the performance of isolated spoken Malay speech recognition using Multi-layer Neural Networks," in *IEEE International Symposium*, pp. 182-186, 2010.

[10] G. Huang and V. Lee, “The ASR technique for meal service robot, ” in *IECON 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 3317-3322, November 7-10, 2011.

[11] M. Anjugam and M. Kavitha, “Design and Implementation of Voice Control System for Wireless Home Automation Networks,” in *International Conference on Computing and Control Engineering (ICCCE)*, pp 1- 6, April 12 - 13, 2012.

[12] B. Lecouteux, M. Vacher, F. Portet, “Distant Speech Recognition for Home Automation: Preliminary Experimental Results in a Smart Home,” in *6th Conference Speech Technology and Human-Computer Dialogue (SPeD)*, pp . 1-10, May 18 -21, 2011.

[13] L. Tsou, “Manual Wheelchair Automator: Design of Front-end Process for a Speech Recognition System Manual Wheelchair Automator,” in *Encyclopedia of Digital Commons@McMaster*, 2009.

[14] B. Juang and L. Rabiner, “Automatic speech recognition–a brief history of the technology development,” in *Elsevier Encyclopedia of Language and Linguistics second edition*, 2005.

[15] F. Rosdi, “Isolated Speech Recognition using Hidden Markov Model,” Degree of Master Dissertation, University of Malaya Kuala Lumpur, May 2008.

[16] T. Ming, “Malay continuous Speech Recognition using Continuous Density Hidden Markov Model,” Degree of Master Dissertation University Technology Malaysia, May 2007.

[17] A. Toh, “Feature extraction for robust speech recognition in hostile enviroment,” PHD Dissertation University of Western Australia , 2007.

[18] R. Jones, S. Downey, J. Mason, “Continuous speech recognition using syllables", in *Fifth European Conference on Speech (Eurospeech)* , pp. 1171-1174 , 1997.

[19] Ian McLoughlin, *Applied Speech and Audio Processing*. Cambridge University Press, 2009.

[20] H. Sakoe and S. Chiba, “Dynamic Programming algorithm Optimization for spoken word Recognition,” *IEEE transaction on Acoustic speech and Signal Processing*, pp. 43-49, February 1978.

[21] M. Nilsson and M. Ejnarsson, “Speech Recognition using Hidden Markov Model performance evaluation in noisy environment,” Degree of Master Dissertation Blekinge Institute of Technology, March 2002.

[22] B. Plannerer, “An introduction to speech recognition,” *Encyclopedia of University of Munich Germany*, March 28, 2005.

[23] “Speech,” *https://en.wikipedia.org/wiki/Speech.html*, 14 May 2013.

[24] H. Dole, “Evaluating the effects of automatic speech recognition word accuracy,” Degree of Master Dissertation of Virginia Technology Institute, July 10, 1998.

[25] AG. Adami, “Automatic Speech Recognition: From the Beginning to the Portuguese Language,” in *9th International Conference Computational Processing of the Portugal Language*, April 27-30, 2010.

[26] S. Das, “Speech Recognition Technique: A Review,” *International Journal of Engineering Research and Application (IJERA)* , Vol. 2, No. 3, pp. 2071-2082, May - Jun 2012.

[27]“AutomaticSpeechUnderstanding”,*http://ewh.ieee.org/r10/bombay/news6/AutoSpeechRecog/ASR",* 16 May 2013.

[28] RS. Kurcan, “Isolated Word Recognition From In - Ear Microphone Data using Hidden Markov Model (HMM), ” Degree of Master Dissertation Naval Postgraduate, March 2006.

[29] “Sampling Rate,” *http://en.wikipedia.org/wiki/Sampling_rate.html*, 14 May 2013.

[30] BH. Juang and L. Rabiner, “Hidden Markov Model for Speech Recognition,” in *Technometrics*, Vol. 33, No. 3, pp. 251-272, Augast 1991.

[31] C. Lee and E. Giachin, “Improved acoustic modeling for continuous speech recognition,” *Acoustic, Speech and Signal Processing (ICASSAP),* Vol. 1 , pp. 161-164 , April 14-17, 1991.

[32] S. Lokesh and G. Balakrishna, “Speech Enhancement using Mel - LPC Cepstrum and Vector Quantization,” *European Journal of Scientific Research*, Vol. 73, No 2, pp. 202 - 209, 2012.

[33] J. Wang, J. Wang and M. Mo, “The design of a speech interactivity embedded module and its applications for mobile consumer devices,” *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 2, pp. 870-876 , May 2008.

[34] F. Muzaffar and B. Mohsin, “Dsp implementation of voice recognition using dynamic time warping algorithm,” *Engineering Sciences and Technology, (SCONEST)* , Vol.1, No. 4, pp. 1-7, 27 August 2005.

[35] M. Fallahzadeh and F. Farokh, “A hybrid reliable Algorithm for Speaker Recognition based on improved DTW and VQ by Genetic Algorithm in noisy environment,” in I*nternational Conference on Multimedia and Signal Processing*, pp. 269-273, 2011.

[36] T. Zaharia and S. Segarceanu, “Quantized Dynamic Time Warping (DTW) algorithm,” in *IEEE Conference Communication*, pp. 91 - 94, June 10-12, 2010.

[37] R. Sabah and R, Ainon, “Isolated digit speech recognition in Malay language using neuro-fuzzy approach,” in *third Asia International comference on Modelling and simulation*, pp. 2-6, 2009.

[38] S. Mahmood and M. Abdulrahim, “Hybrid Speech Recognition System based on Coefficient,” *International Conference on Emerging Trends in Computer and Electronics Engineerin*g *(ICETCEE),* No 1, pp 1 - 4, March 24 - 25 2012.

[39] B. Das and R. Parekh, “Recognition of Isolated Words using Features based on LPC, MFCC, ZCR and STE, with Neural Network Classifiers,” *International Journal of Modern Engineering Research*, Vol. 2, No 3, pp 854 - 858, May - June 2012.

[40] G. Zhang, J. Yin and Qian Liu et al, “The fixed-point optimization of mel frequency cepstrum coefficients for speech recognition,” in *The 6th International Forum on Strategic Technology*, pp. 1172 - 1175, 2011.

[41] Z. Yu-Zheng, L. and L. Xiao-ying, “The Application of Adaptively Enhanced Bark Wavelet MFCC and The Introduction of A Novel Noise-robust Speech Recognition System,” in *Proceedings of the International Workshop on Information Security and Application*, pp. 168 - 172, 2009.

[42] S. Suk and H. Kojima, "Voice Activated Appliances for Severely Disabled Persons," in *Speech Recognition Technology and Application*, pp. 1-12, November 2008.

[43] E. Thakur, A. Singla and V. Patil, "Design of Hindi Key Word Recognition System for Home Automation System Using MFCC and DTW, " *International Journal Of Advanced Engineering Science and Technologies*, Vol. 11, No. 1 pp. 177-182, 2011.

[44] P. Cernys and V. Kubilius, " Intelligent control of the lift model," in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 428-431, September 8-10 2003.

[45] G. Pacnik, K. Benkiс and B. Brecko, "Voice operated intelligent wheelchair-VOIC," in *IEEE Industrial Electronics (ISIE)*, pp. 1221-1226, June 20-23, 2005.

[46] K. Peng, H. Cai and Y. Zhang, "Linear histogram equalization in the acoustic feature domain for speech recognition over Bluetooth™ channels, " *4th International Conference on Mobile Technology, Applications, and Systems*, Vol. 7, No. 7, pp. 427-430, 2007.

[47] X. Zeng, A. Fapojuwo and R. Davies, "Design and Performance Evaluation of Voice Activated Wireless Home Devices," *Consumer Electronics IEEE Transactions*, Vol. 52, No. 3, pp. 983-989, Augast 2006.

[48] R.Bajcsy and S.Kovacic, "Multiresolution Elastic Matching, " *Computer Vision Graphics and Image Processing*, Vol. 46, No. 1, pp. 1-21, 1989.

[49] H. Boulard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publisher, 1994.

[50] T. Tan and B. Renaivo-Malancon, "Malay Grapheme to Phoneme Tool for Automatic Speech Recognition," in *Third International Workshop on Malay* , pp. 1-6, 2009.

[51] MJ. Yap, SJ. Rickard S. Faizal and S. Jalil, "The Malay Lexicon Poject: A database of lexical statistic for 9,592 words," *Behavior Research Methods*, Vol. 42, No. 4, pp. 992-1003, November 2010.

[52] N. Mat Awal, K. Abu Bakar and N. Abdul Hamid, *Morphological Differences Between Bahasa Melayu And English: Constraints In Students' Understanding*, University Kebangsaan Malaysia, 2008.

[53] Z. Razak, N. Ibrahim and E. Tamil et al, "Quranic Verse Recitation Feature Extraction Using Mel-Frequency Cepstral Coefficient (MFCC)," *International Journal of Computer science and Network Security (IJCSNS)*, Vol. 8, No. 8, March 2008.

[54] K. Aida–Zade, C. Ardil and S. Rustamov, "Investigation of combined use of MFCC and LPC Features in Speech Recognition Systems, " in *World Academy of Science, Engineering and Technology* , pp. 74-80, 2006.

[55] F. Zheng, G. Zhang and G. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, Vol. 16, No. 6, pp. 1-7, September 2001.

[56] X. Anguera, R. Macrae and N. Oliver, "Partial Sequence Matching using an Unbounded Dynamic Time Warping Agorithms," in *Acoustics Speech and Signal Processing (ICASSP) IEEE International Conference* , pp. 3582 - 3585 , March 14-19 2010.

[57] A. Acero and X. Huang, "Augmented Cepstral Normalization for Robust Speech Recognition," in *Proc. of IEEE Automatic Speech Recognition*, 1995.

[58] D. Feifei and H. Qizhi, "Speech Endpoint Detection Based on Improved Cepstral Mean Subtraction," in *International Conference on Intelligent System Design and Engineering Apllication*, pp. 1121-1124, 2012.

[59] A. Fazel and S. Chakrabartty, "An Overview of Statistical Pattern Recognition Techniques for Speaker Vertification," *Circuits and Systems Magazine, IEEE,* Vol. 11, No. 2, pp. 62-81, June 2011.

[60] W. Hong, and P. Jin, "Modified MFCC for robust spaker recognition," *Intelligent Computing and Intelligent Systems (ICIS) IEEE International Conference* , Vol. 1, pp 276 - 279 , October 29-31 2010.

[61] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space," in Intelligent Data Analysis-IOS Press, pp 70-80 , 2007.

[62] C. Fang, "From Dynamic time warping (DTW) to Hidden Markov Model (HMM)," University of Cincinnati, Technical Report, 2009.

[63] FK. Soong, AE. Rosenberg, BH. Juang and L. Rabiner, "A Vector Quantization Approach to Speaker Recognition," *AT&T Technical Journal*, Vol. 66, No. 2, pp. 14-26, April 29, 2014.

[64] TJ. Watson, "Robust Speech Recognition," in *Encyclopedia of Research Center Fracarro Radio industries*, April 9,2001.

[65] D. Colton and B. Hawaii, "Automatic Speech Recognition Tutorial," in *Encyclopedia*, June 17, 2003.

[66] A. Balla and S. Khaparkar, "Perfromance Improvement of Speaker Recognition System," *International Journal of advanced Research in Computer Science and software Engineering*, Vol. 2, No. 3, March 2012.

[67] T. Kinnunen, R. Saeidi and J. Sandberg et al., "What Else is New Than the Hamming Window? Robust MFCCs for Speaker Recognition via Multitapering," in *InterSpeech Conferenc*e, pp 2734 - 2737 , September 2010.

[68] S. Nisar, M. Ali Khan, M. Usman, "IRIS Recognition using Mel Frequency Cepstral Coefficient," *International Journal of Engineering Research*, Vol. 3, No. 2 pp. 100 - 103 , February 2014.

[69] S. Gupta, J. Jaafar, W. Ahmad and A. Bansal, "Feature Extraction using MFCC, " *Signal and Image Processing: An International Journal (SIPIJ)*, Vol. 4, No. 3, Augast 2013.

[70] A. Bala, "Voice Command Recognition system Based on MFCC and DTW, " *International Journal of Engineering Science and Technology*, Vol. 2, No. 12 pp. 7335 - 7342, 2010.

[71] M. Kesarkar, "Feature Extraction for Speech Recognition," in *M. Tech. Credit Seminar Report, Electronic System Group*, pp. 1-12, November 2003.

[72] M. Anusuya and S. Katti, "Speech recognition by machine: A review," *International Journal of Computer Science and Information Security*, Vol. 6, No. 3, pp. 191 - 205, 2009.

[73] M. Borosh, W. Eckert and F. Galwitz, "Towards understanding spontaneous speech: Word accuracy vs. concept accuracy," *Spoken Language, Proceedings, Fourth International Conference (ICSLP)*, Vol. 2, pp. 1009 - 1012, October 3-6 1996.

[74] B. Malancon, "Computational analysis of Affixed words in Malay Language, " in *Encyclopedia of University Science Malaysia*, 2004.

[75] Li Zhang, "A syllables-Based, Pseudo-Articulatory Approach to Speech Recongnition," in *Encyclopedia of School of Computer Science University of Birmingham* , September 2004.

[76] I. Bazzi, "Modelling Out-of-Vocabulary Words for Robust Speech Recognition," in *Encyclopedia of Massachusetts Institute Of Technology* , June 2002.

[77] S. Mahon, "English Vowel Digraph and their History," in *Encyclopedia of Linguistic LIN 4970*, 20 December 2011.

[78] Z. Hanim and A. Khalifa, "Towards Designing A High Intelligibility Rule Based Standard Malay Text-To-Speech Synthesis System," *Proceeding of the International Conference on Computer and Communication Engineering* , Vol. 31, No. 10, June 2008.

[79] M. Salam, D. Mohamad and S. Salleh, "Malay Isolated Speech Recognition Using Neural Network: A Work in Finding Number of Hidden Nodes and Learning Parameters," *The International Arab Journal of Information Technology*, Vol. 8, No. 4, pp 364-372, 15-17 October 2011.

[80] W. Abdulla, D. Chow and G, Sin et al., "Cross-words reference template for DTW-based speech recognition systems," *Conference on Convergent Technologies for the Asia-Pacific Region (TENCON)*, Vol. 4, pp. 1576-1579, October 15-17 2003.

[81] NH. Samsudin, S. Tiun and TE. Kong, "A simple Malay Speech Synthesizer using Syllable Concatenation Approach, " in *Proceeding National Computer Science Postgraduate Colloquium*, 2005.

[82] M. Donohue, "Condition on Stress in Varieties of Malay/Indonesia," in *The Eleventh International Symposium on Malay/ Indonesian Linguistics (ISMIL)* , Augast 6-8 2007.

[83] LJ. Brinton, *The Structure of Modern English A Linguistic Introduction*. John Benjamin Publishing, 2000.

[84] L. Olivier, "Designing a smart home environment using wireless sensor networking of everyday objects," Degree of Master Dissertation, UMEA University, November 27 2008.

[85] N. Souto, "Building language models for continuous speech recognition systems," in *Springer-Verlag Berlin*, pp. 101-120, 2002.

[86] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[87] C. Yee and A. Ahmad, "Malay Language Text-Independent Speaker verification using NN-MLP classifier With MFCC," in *International Conference on Electronic Design* , pp. 1-5, December 1-3 2008.

[88] Andrea audio test lab, "Andrea Pure Audio BT-200 Noise Cancelling Bluetooth Headset Performance Comparative Testing,", Technical Report, November 26, 2008.

[89] Whitepaper, "Understanding the difference between 900MHz and 2.4GHz," in *Encyclopedia of Plantronic Inc*. , 2003.

[90] Whitepaper, "Wi-fi and Bluetooth Interference Issue," in *Encyclopedia of Hp Invent Inc.* , January 2002.

[91] A. Ouzounov, "Cepstral Features and Text Dependent Speaker Identification a Comparative Study," *Cybernatics and Information Technologies*, Vol. 10, No. 1, 2010.

[92] T. Rath, and R. Manmatha, "Word Image Matching Using Dynamic Time Warping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. II-521-II-527, 2003.

LIST OF PUBLICATIONS

[1] L. Muda, M. Begam I. Elamvazuthi, , "Voice Recognition Algorithms using Mel Frequency Cepstral coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques," *Journal of Computing*, vol. 2, no. 3,pp 138-143, 2010.

[2] L. Muda, M. Begam I. Elamvazuthi, , M. L. M. Zain, and R.Rashid, "Determination of Speech Recognition Accuracy using DTW and MFCC Techniques," in *International Conference on Robotic Automation System (ICORAS)*, 2011.

[3] L. Muda, M.Begam and I. Elamvazuthi, "Voice Recognition Algorithms in the Development of a Universal Remote Control," in *Second International Conference and Workshops on Basic and Applied Sciences and Regional Annual Fundamental Science Seminar 2009 (ICORAFSS 2009)*, 2009.

[4] L. Muda,M.Begam and I. Elamvazuthi, , "Development of wireless voice activated remote control for electronics gadget," in *National Postgraduate Conference on Engineering, Science and Technology*, 2009.

APPENDIX A: MICROPHONE

This appendix provides specification of wired and wireless microphone.

1. Wired Microphone

Figure A.1 Sonic Gear Wired Microphone

Additional Information

Features

PC headset with noise-cancellation mic. Suitable for multiplayer gaming. Speech recongnition, music listening. Video Chat

*3.5mm Gold-Plated Jack
*Extra Pair of High Quality Cushion
*2m Cord Length
*Clear Bass
*Multiplayer Internet Gaming

Warranty

- 1-Year Distributor Limited Warranty

Specifications

Specifications * View Official Website *	
Model	SonicGear HS 555 Headset
Tech Specs	Microphone : Electret, Directional Speaker Dimension: 40mm Impedence: 32 Ohms +/- 15% Sensitivity: 105 App. 4dB SPL/0.179 V at 1kHz Frequency Response: 20 - 20kHz Cord Length: 2.5m Operational Temperature Range: - 20 - 55°C Storage Temperature: - 30 - 70°C

Figure A.2 Sonic Gear Specification

2. Wireless Microphone

Figure A.3 Sony PS3 Wireless Headset 5.1

Specifications

Power source	DC 3.7 V: Built-in lithium-ion rechargeable battery
Battery capacity	570 mAh
Operating environment temperature	5°C - 35°C / 41°F - 95°F
Dimensions (w/h/d)	Wireless stereo headset: Approx. 186 × 197 × 95 mm (7.3 × 7.8 × 3.7 in.) Wireless adaptor: Approx. 17.4 × 7 × 64 mm (0.7 × 0.3 × 2.5 in.)
Weight	Wireless stereo headset: Approx. 275 g Wireless adaptor: Approx. 7 g
Communication system	2.4 GHz RF
Maximum communication range	Approx. 10 m*1
Use time when fully charged	Up to 7.5 hours*2
Contents	Wireless stereo headset (1) / Wireless adaptor (1) / Instruction manual (1)

*1 Actual communication range varies depending on factors such as obstacles between the headset and wireless adaptor, magnetic fields from electronics (such as a microwave oven), static electricity, antenna performance, and the operating system and software applications in use. Depending on the operating environment, reception may be interrupted.

*2 Actual use time varies depending on factors such as the battery charge and ambient temperature.

APPENDIX B: MATLAB CODE FOR DTW

This appendix provides a matlab code for DTW technique.

1) Start.m

```
clear all;
 close all;
 clc;
disp (' ');
disp (‘Dynamic Time Warping (DTW) for Home Automation Speech Recognition ');
disp ('                    by Lindasalwa Muda');
Disp (‘------------------------------------------------------------');
Select=input ('    >> Record or Open wave [Default=Enter=Open]? (R/O)','s');
switch Select
   case {'R','r'}
      Option='Record_mode';                          % press r to record new speech
   case {'O', 'o'}
      Option='Open_mode';                            % press o to open test data
   otherwise
      Option='Default_mode';
end
Main_DTW (Option);
clear Select Option
```

2) OpenWave.m

```
function [TestWave, Fs] =OpenWave
file=input ('    >> Wave file name [Default:"1_ChangeChannel.wav"]:','s');
        if strcmp(file,'\n') ==0
file='G:
\Recording_2013\Malay_Modi2013\Wireless\Test\Yasmin\KuatkanSuara.wav'; disp ('
Default: 1_ChangeChannel.wav');      % directory path training or save template
        elseif findstr (file,'.')==''
           file=strcat(file,'.wav');
        end
        [TestWave, Fs, NBits] = wavread (file);
         Wavplay (TestWave, Fs, 'sync');
        % ===== Plot recorded waveform
        %hold on
       % plot (TestWave,'b--');
        %title ('Recorded wave file');
        %axis ([0 Duration min (Out) max (Out)]);
       % grid on;
        %hold off
         % ==== Save recorded file
         FileName = 'Test.wav';
         wavwrite (TestWave, Fs, 8, Filename);
         fprintf ('    >> The file is save to "%s"\n', FileName);
```

3) DTW_Create_Template.m

```
% Dynamic Time Warping (DTW)
 % extracting features of Templates and save them.
 clear all;
 close all;
clc;
 %Path='D:\MYProject\Voice_DTW-Versi-Linda\Data\';
Template_List=['1_Buka','1_Tutup','1_KuatkanSuara','1_PerlahankanSuara','1_Tukar
Siaran','2_Buka','2_Tutup','2_KuatkanSuara','2_PerlahankanSuara','2_TukarSiaran','3_
Buka','3_Tutup','3_KuatkanSuara','3_PerlahankanSuara','3_TukarSiaran"4_Buka','4_T
utup','4_KuatkanSuara','4_PerlahankanSuara','4_TukarSiaran','5_Buka','5_Tutup','5_K
uatkanSuara','5_PerlahankanSuara','5_TukarSiaran','6_Buka','6_Tutup','6_KuatkanSua
ra','6_PerlahankanSuara','6_TukarSiaran','7_Buka','7_Tutup','7_KuatkanSuara','7_Perl
ahankanSuara','7_TukarSiaran','8_Buka','8_Tutup','8_KuatkanSuara','8_PerlahankanS
uara','8_TukarSiaran','9_Buka','9_Tutup','9_KuatkanSuara','9_PerlahankanSuara','9_T
ukarSiaran','10_Buka','10_Tutup','10_KuatkanSuara','10_PerlahankanSuara','10_Tuka
rSiaran','11_Buka','11_Tutup','11_KuatkanSuara','11_PerlahankanSuara','11_TukarSia
ran','12_Buka','12_Tutup','12_KuatkanSuara','12_PerlahankanSuara','12_TukarSiaran'
];

 Template_MFCC_Features_zero= CMS_Normalization (Feature_Extruction
([Path,'1_Buka.wav']));
 Template_MFCC_Features_one= CMS_Normalization (Feature_Extruction
([Path,'1_Tutup.wav']));
 Template_MFCC_Features_two= CMS_Normalization (Feature_Extruction
([Path,'1_KuatkanSuara.wav']));
 Template_MFCC_Features_three= CMS_Normalization (Feature_Extruction
([Path,'1_PerlahankanSuara.wav']) ;
 Template_MFCC_Features_four= CMS_Normalization (Feature_Extruction
([Path,'1_TukarSiaran.wav']));
 Template_MFCC_Features_five= CMS_Normalization (Feature_Extruction
([Path,'2_Buka.wav']));
 Template_MFCC_Features_six= CMS_Normalization (Feature_Extruction
([Path,'2_Tutup.wav']));
 Template_MFCC_Features_seven= CMS_Normalization (Feature_Extruction
([Path,'2_KuatkanSuara.wav']));
 Template_MFCC_Features_eight= CMS_Normalization (Feature_Extruction
([Path,'2_PerlahankanSuara.wav']));
 Template_MFCC_Features_nine= CMS_Normalization (Feature_Extruction
([Path,'2_TukarSiaran.wav']));
 Template_MFCC_Features_ten= CMS_Normalization (Feature_Extruction
([Path,'3_Buka.wav']));
 Template_MFCC_Features_oneone= CMS_Normalization (Feature_Extruction
([Path,'3_Tutup.wav']));
 Template_MFCC_Features_onetwo= CMS_Normalization (Feature_Extruction
([Path,'3_KuatkanSuara.wav']));
 Template_MFCC_Features_onethree= CMS_Normalization (Feature_Extruction
([Path,'3_PerlahankanSuara.wav']));
```

```
 Template_MFCC_Features_onefour= CMS_Normalization (Feature_Extruction
([Path,'3_TukarSiaran.wav']));
 Template_MFCC_Features_onefive= CMS_Normalization (Feature_Extruction
([Path,'4_Buka.wav']));
 Template_MFCC_Features_onesix= CMS_Normalization (Feature_Extruction
([Path,'4_Tutup.wav']));
 Template_MFCC_Features_oneseven= CMS_Normalization (Feature_Extruction
([Path,'4_KuatkanSuara.wav']));
 Template_MFCC_Features_oneeight= CMS_Normalization (Feature_Extruction
([Path,'4_PerlahankanSuara.wav']));
 Template_MFCC_Features_onenine=
CMS_Normalization(Feature_Extruction([Path,'4_TukarSiaran.wav']));
 Template_MFCC_Features_twoone=
CMS_Normalization(Feature_Extruction([Path,'5_Buka.wav']));
 Template_MFCC_Features_twotwo=
CMS_Normalization(Feature_Extruction([Path,'5_Tutup.wav']));
 Template_MFCC_Features_twothree=
CMS_Normalization(Feature_Extruction([Path,'5_KuatkanSuara.wav']));
 Template_MFCC_Features_twofour=
CMS_Normalization(Feature_Extruction([Path,'5_PerlahankanSuara.wav']));
 Template_MFCC_Features_twofive=
CMS_Normalization(Feature_Extruction([Path,'5_TukarSiaran.wav']));
 Template_MFCC_Features_twosix=
CMS_Normalization(Feature_Extruction([Path,'6_Buka.wav']));
 Template_MFCC_Features_twoseven=
CMS_Normalization(Feature_Extruction([Path,'6_Tutup.wav']));
 Template_MFCC_Features_twoeight=
CMS_Normalization(Feature_Extruction([Path,'6_KuatkanSuara.wav']));
 Template_MFCC_Features_twonine=
CMS_Normalization(Feature_Extruction([Path,'6_PerlahankanSuara.wav']));
 Template_MFCC_Features_threeone=
CMS_Normalization(Feature_Extruction([Path,'6_TukarSiaran.wav']));
 Template_MFCC_Features_threetwo=
CMS_Normalization(Feature_Extruction([Path,'7_Buka.wav']));
 Template_MFCC_Features_threethree=
CMS_Normalization(Feature_Extruction([Path,'7_Tutup.wav']));
 Template_MFCC_Features_threefour=
CMS_Normalization(Feature_Extruction([Path,'7_KuatkanSuara.wav']));
 Template_MFCC_Features_threefive=
CMS_Normalization(Feature_Extruction([Path,'7_PerlahankanSuara.wav']));
 Template_MFCC_Features_threesix=
CMS_Normalization(Feature_Extruction([Path,'7_TukarSiaran.wav']));
 Template_MFCC_Features_threeseven=
CMS_Normalization(Feature_Extruction([Path,'8_Buka.wav']));
 Template_MFCC_Features_threeeight=
CMS_Normalization(Feature_Extruction([Path,'8_Tutup.wav']));
 Template_MFCC_Features_threenine=
CMS_Normalization(Feature_Extruction([Path,'8_KuatkanSuara.wav']));
```

```
 Template_MFCC_Features_fourone=
CMS_Normalization(Feature_Extruction([Path,'8_PerlahankanSuara.wav']));
 Template_MFCC_Features_fourtwo=
CMS_Normalization(Feature_Extruction([Path,'8_TukarSiaran.wav']));
 Template_MFCC_Features_fourthree=
CMS_Normalization(Feature_Extruction([Path,'9_Buka.wav']));
 Template_MFCC_Features_fourfour=
CMS_Normalization(Feature_Extruction([Path,'9_Tutup.wav']));
 Template_MFCC_Features_fourfive=
CMS_Normalization(Feature_Extruction([Path,'9_KuatkanSuara.wav']));
 Template_MFCC_Features_foursix=
CMS_Normalization(Feature_Extruction([Path,'9_PerlahankanSuara.wav']));
 Template_MFCC_Features_fourseven=
CMS_Normalization(Feature_Extruction([Path,'9_TukarSiaran.wav']));
 Template_MFCC_Features_foureight=
CMS_Normalization(Feature_Extruction([Path,'10_Buka.wav']));
 Template_MFCC_Features_fournine=
CMS_Normalization(Feature_Extruction([Path,'10_Tutup.wav']));
 Template_MFCC_Features_fiveone=
CMS_Normalization(Feature_Extruction([Path,'10_KuatkanSuara.wav']));
 Template_MFCC_Features_fivetwo=
CMS_Normalization(Feature_Extruction([Path,'10_PerlahankanSuara.wav']));
 Template_MFCC_Features_fivethree=
CMS_Normalization(Feature_Extruction([Path,'10_TukarSiaran.wav']));
 Template_MFCC_Features_fivefour=
CMS_Normalization(Feature_Extruction([Path,'11_Buka.wav']));
 Template_MFCC_Features_fivefive=
CMS_Normalization(Feature_Extruction([Path,'11_Tutup.wav']));
 Template_MFCC_Features_fivesix=
CMS_Normalization(Feature_Extruction([Path,'11_KuatkanSuara.wav']));
 Template_MFCC_Features_fiveseven=
CMS_Normalization(Feature_Extruction([Path,'11_PerlahankanSuara.wav']));
 Template_MFCC_Features_fiveeight=
CMS_Normalization(Feature_Extruction([Path,'11_TukarSiaran.wav']));
 Template_MFCC_Features_fivenine=
CMS_Normalization(Feature_Extruction([Path,'12_Buka.wav']));
 Template_MFCC_Features_sixone=
CMS_Normalization(Feature_Extruction([Path,'12_Tutup.wav']));
 Template_MFCC_Features_sixtwo=
CMS_Normalization(Feature_Extruction([Path,'12_KuatkanSuara.wav']));
 Template_MFCC_Features_sixthree=
CMS_Normalization(Feature_Extruction([Path,'12_PerlahankanSuara.wav']));
 Template_MFCC_Features_sixfour=
CMS_Normalization(Feature_Extruction([Path,'12_TukarSiaran.wav']));
 %save Templates.mat
 save Templates_data1.mat

 clear path
```

4) SelectNextTemplate.m

```
function [Temp_F, Temp_N] =SelectNextTemplate (No) %function to display recognize
word

 % Select the next template and return its name (Temp_N) and feature vectors (Temp_F)

   load Templates_data1.mat;

   Switch (No)
      Case {1}
         Temp_F=Template_MFCC_Features_zero;
         Temp_N='1_Buka';
      case {2}
         Temp_F=Template_MFCC_Features_one;
         Temp_N='1_Tutup';
      case {3}
         Temp_F=Template_MFCC_Features_two;
         Temp_N='1_KuatkanSuara';
      case {4}
         Temp_F=Template_MFCC_Features_three;
         Temp_N='1_PerlahankanSuara';
      case {5}
         Temp_F=Template_MFCC_Features_four;
         Temp_N='1_TukarSiaran';
      case {6}
         Temp_F=Template_MFCC_Features_five;
         Temp_N='2_Buka';
      case {7}
         Temp_F=Template_MFCC_Features_six;
         Temp_N='2_Tutup';
      case {8}
         Temp_F=Template_MFCC_Features_seven;
         Temp_N='2_KuatkanSuara';
      case {9}
         Temp_F=Template_MFCC_Features_eight;
         Temp_N='2_PerlahankanSuara';
      case {10}
        Temp_F=Template_MFCC_Features_nine;
         Temp_N='2_TukarSiaran';
      case {11}
          Temp_F=Template_MFCC_Features_ten;
          Temp_N='3_Buka';
      case {12}
          Temp_F=Template_MFCC_Features_oneone;
          Temp_N='3_Tutup';
      case {13}
          Temp_F=Template_MFCC_Features_onetwo;
```

```
    Temp_N='3_KuatkanSuara';
case {14}
    Temp_F=Template_MFCC_Features_onethree;
    Temp_N='3_PerlahankanSuara';
case {15}
    Temp_F=Template_MFCC_Features_onefour;
    Temp_N='3_TukarSiaran';
 case {16}
    Temp_F=Template_MFCC_Features_onefive;
    Temp_N='4_Buka';
 case {17}
    Temp_F=Template_MFCC_Features_onesix;
    Temp_N='4_Tutup';
 case {18}
    Temp_F=Template_MFCC_Features_oneseven;
    Temp_N='4_KuatkanSuara';
 case {19}
    Temp_F=Template_MFCC_Features_oneeight;
    Temp_N='4_PerlahankanSuara';
 case {20}
    Temp_F=Template_MFCC_Features_onenine;
    Temp_N='4_TukarSiaran';
case {21}
    Temp_F=Template_MFCC_Features_twoone;
    Temp_N='5_Buka';
case {22}
    Temp_F=Template_MFCC_Features_twotwo;
   Temp_N='5_Tutup';
 case {23}
   Temp_F=Template_MFCC_Features_twothree;
    Temp_N='5_KuatkanSuara';
 case {24}
    Temp_F=Template_MFCC_Features_twofour;
    Temp_N='5_PerlahankanSuara';
 case {25}
    Temp_F=Template_MFCC_Features_twofive;
   Temp_N='5_TukarSiaran';
 case {26}
    Temp_F=Template_MFCC_Features_twosix;
    Temp_N='6_Buka';
case {27}
    Temp_F=Template_MFCC_Features_twoseven;
   Temp_N='6_Tutup';
 case {28}
   Temp_F=Template_MFCC_Features_twoeight;
    Temp_N='6_KuatkanSuara';
 case {29}
    Temp_F=Template_MFCC_Features_twonine;
    Temp_N='6_PerlahankanSuara';
```

```
 case {30}
    Temp_F=Template_MFCC_Features_threeone;
   Temp_N='6_TukarSiaran';
 case {31}
    Temp_F=Template_MFCC_Features_threetwo;
    Temp_N='7_Buka';
case {32}
    Temp_F=Template_MFCC_Features_threethree;
   Temp_N='7_Tutup';
 case {33}
   Temp_F=Template_MFCC_Features_threefour;
    Temp_N='7_KuatkanSuara';
 case {34}
    Temp_F=Template_MFCC_Features_threefive;
    Temp_N='7_PerlahankanSuara';
 case {35}
    Temp_F=Template_MFCC_Features_threesix;
   Temp_N='7_TukarSiaran';
case {36}
    Temp_F=Template_MFCC_Features_threeseven;
    Temp_N='8_Buka';
case {37}
    Temp_F=Template_MFCC_Features_threeeight;
   Temp_N='8_Tutup';
 case {38}
   Temp_F=Template_MFCC_Features_threenine;
    Temp_N='8_KuatkanSuara';
 case {39}
    Temp_F=Template_MFCC_Features_fourone;
    Temp_N='8_PerlahankanSuara';
 case {40}
    Temp_F=Template_MFCC_Features_fourtwo;
   Temp_N='8_TukarSiaran';
case {41}
    Temp_F=Template_MFCC_Features_fourthree;
   Temp_N='9_Buka';
case {42}
    Temp_F=Template_MFCC_Features_fourfour;
   Temp_N='9_Tutup';
case {43}
    Temp_F=Template_MFCC_Features_fourfive;
   Temp_N='9_KuatkanSuara';
case {44}
    Temp_F=Template_MFCC_Features_foursix;
   Temp_N='9_PerlahankanSuara';
case {45}
    Temp_F=Template_MFCC_Features_fourseven;
   Temp_N='9_TukarSiaran';
```

```
    case {46}
        Temp_F=Template_MFCC_Features_foureight;
       Temp_N='10_Buka';
    case {47}
        Temp_F=Template_MFCC_Features_fournine;
       Temp_N='10_Tutup';
    case {48}
        Temp_F=Template_MFCC_Features_fiveone;
       Temp_N='10_KuatkanSuara';
    case {49}
        Temp_F=Template_MFCC_Features_fivetwo;
       Temp_N='10_PerlahankanSuara';
    case {50}
        Temp_F=Template_MFCC_Features_fivethree;
        Temp_N='10_TukarSiaran';
    case {51}
        Temp_F=Template_MFCC_Features_fivefour;
       Temp_N='11_Buka';
    case {52}
        Temp_F=Template_MFCC_Features_fivefive;
       Temp_N='11_Tutup';
    case {53}
        Temp_F=Template_MFCC_Features_fivesix;
       Temp_N='11_KuatkanSuara';
    case {54}
        Temp_F=Template_MFCC_Features_fiveseven;
         Temp_N='11_PerlahankanSuara';
    case {55}
        Temp_F=Template_MFCC_Features_fiveeight;
        Temp_N='11_TukarSiaran';
    case {56}
        Temp_F=Template_MFCC_Features_fivenine;
       Temp_N='12_Buka';
    case {57}
        Temp_F=Template_MFCC_Features_sixone;
       Temp_N='12_Tutup'
    case {58}
        Temp_F=Template_MFCC_Features_sixtwo;
       Temp_N='12_KuatkanSuara';
    case {59}
        Temp_F=Template_MFCC_Features_sixthree;
         Temp_N='12_PerlahankanSuara';
    case {60}
        Temp_F=Template_MFCC_Features_sixfour;
       Temp_N='12_TukarSiaran';
    otherwise
       error;
end
```

5) Main_FeatureExtraction.m

```
% Test Feature_Extraction function
clear all;
close all;
clc
InWave='1_ChangeChannel.wav'; % default wave file in Fs Hz, with Duration
                              seconds
Fs=8000;
Duration=2;                   % in seconds

MFCC_Featurs= Feature_Extruction (InWave, Fs);   % MFCC only
CMS_MFCC=CMS_Normalization(MFCC_Featurs); % MFCCCMS
```

6) Feature_Extraction.m

```
%       ***********************************************
%       ******     Feature extraction          *******
%       ***********************************************
function Featurs= Feature_Extruction(InputWave,Fs)
   % Featurs= Feature_Extruction (InputWave);
   %    Return 39 MFCC feature vectors of InputWave

   if nargin<1
      disp('Error: in Feature_Extruction, no wave file.');
   end
   if isstr(InputWave),
      [InputWave,Fs,NBits] = wavread(InputWave);
   elseif nargin==1
      Fs=8000;
   end

   % ====== Set Parameters=====================================
   %          Frame size: N (ms), Overlapping region is M (ms)
   %          Generally , M = (1/2)*N , which N = 24.
   FrameSize_ms = 24;      % Ex. N=32 = (256/8000)*1000, each frame has 256 points.
   Overlap_ms = (1/2)*FrameSize_ms;   % overlap size
   FrameSize = round (FrameSize_ms*Fs/1000); % 256
   Overlap = round (Overlap_ms*Fs/1000); % 86
   % Triangular Band Filter parameters: StartFreq, CenterFreq, StopFreq. (20 Bank filters)
    StartFreq=[1   3   5   7   9   11  13  15  17  19  23  27  31  35  40  46  55  61  70
81];   %Start
    CenterFreq=[3  5   7   9   11  13  15  17  19  21  27  31  35  40  46  55  61  70  81
93];   %Center
    StopFreq=[5    7   9   11  13  15  17  19  21  23  31  35  40  46  55  61  70  81  93
108];  %End
Threshold = 0.0001;   % for energy test ==> remove frames with energy bellow this
                        amount.
```

```
% ====== Step 1: Pre-emphasis ============================
InputWave = filter([1, -0.95], 1, InputWave);  % emphasize the signal
figure(1)
plot (InputWave,'-b')

% ====== Step 2: Windowing & overlapping ==================
Frame = buffer2 (InputWave, FrameSize, Overlap);
normalize_coff = 10;
energy = sum (Frame. ^2)/FrameSize;
index = find(energy < Threshold);
energy(index) = [];
logEnergy = 10*log10(energy)/normalize_coff;
Frame (:, index) = [];                                  % Remove empty frames
Featurs = [];
for i = 1:size(Frame, 2);   % size(Frame, 2)=No_of_Frames

    % ====== Step 3: Hamming window=======================
    WindowedFrame = hamming (FrameSize).*Frame (:, i);

    % ====== Step 4: FFT: fast Fourier transform.
    %     using FFT function to calculate.
    %    Compute square of real part and imaginary part

    FFT_Frame = abs (fft (WindowedFrame));

    % ====== Step 5: Triangular band pass filter.
    % using user defined function triBandFilter (fftFrame {i}).

    No_of_FilterBanks = 20; %No_of_FilterBanks
   tbfCoef = TriBandFilter (FFT_Frame, No_of_FilterBanks, StartFreq, CenterFreq, StopFreq);

    % ====== Step 6: Logarithm.
    tbfCoef = log(tbfCoef.^2);

    % ====== Step 7: DCT: Discrete Cosine Transform.
    % using DCT to get L order mel-scale-cepstrum parameters.

    No_of_Featurs = 12;      % generally No_of_Featurs is 12.
    Cepstrums = Mel_Cepstrum2(No_of_Featurs,No_of_FilterBanks,tbfCoef);
    Featurs = [Featurs Cepstrums'];
     end;
Featurs = [Featurs; logEnergy]; % MFCC 39 Coefficient delta delta energy

%=========compute delta energy and delta cepstrum============
.
Delta_window = 2;            %Calculate delta cepstrum and delta log energy
D_Featurs = DeltaFeature(Delta_window, Featurs);
   %=========compute delta-delta energy and delta cepstrum============
%calculate delta-delta cepstrum and delta log energy
```

```
   %Combine them with previous features, get 39 order Features.

   %Delta_window = 2;
   %D_d_Features = Delta_DeltaFeature(Delta_window, Features);
   % or
   D_d_Features = DeltaFeature(Delta_window, D_Features);

   %===== Combine cepstrum,delta and delta-delta
    Features = [Features ; D_Features ; D_d_Features];  % 39 features

%============================ Sub function ================

%       ***********************************************
%       ******    Triangular Band Filter       *******
%       ***********************************************

function tbfCoef = TriBandFilter (fftFrame, P, StartFreq, CenterFreq, StopFreq)
   %the function is triangular band pass filter

   for i = 1 : P,
      % Compute the slope of left side of triangular band pass filter
     for j = StartFreq(i) : CenterFreq(i),
       filtmag(j) = (j-StartFreq(i))/(CenterFreq(i)-StartFreq(i));
     end;
      % Compute the slope of right side of triangular band pass filter
     for j = CenterFreq(i)+1: StopFreq(i),
       filtmag(j) = 1-(j-CenterFreq(i))/(StopFreq(i)-CenterFreq(i));
     end;
     tbfCoef(i) =
sum(fftFrame(StartFreq(i):StopFreq(i)).*filtmag(StartFreq(i):StopFreq(i))');
   end;


%       ***********************************************
%       ******     Mel-scale cepstrums         *******
%       ***********************************************
function Cepstrum = Mel_Cepstrum2(L,P,tbfCoef)
   %compute mel-scale cepstrum, L should be 12 at most part.
   for i=1:L,
      coef = cos((pi/P)*i*(linspace(1,P,P)-0.5))';
      Cepstrum(i) = sum(coef.*tbfCoef');
   end;

%       ***********************************************
%       ******       Delta cepstrums          *******
%       ***********************************************

function D_Features = DeltaFeature (delta_window, Features)          % Compute delta
cepstrum
```

and delta log energy

```
   rows = size (Featurs, 1);
   cols  = size(Featurs,2);
   temp  = [zeros(rows,delta_window) Featurs zeros(rows,delta_window)];
   D_Featurs = zeros(rows,cols);
   denominator = sum([1:delta_window].^2)*2;
   for i = 1+delta_window : cols+delta_window,
     subtrahend = 0;
     minuend    = 0;
     for j = 1 : delta_window,
       subtrahend = subtrahend + temp(:,i+j)*j;
       minuend    = minuend + temp(:,i-j)*(-j);
     end;
     D_Featurs(:,i-delta_window) = (subtrahend - minuend)/denominator;
   end;
 Featurs = [Featurs; temp2];

%       ************************************************
%       ******      Delta-Delta cepstrums       *******
%       ************************************************

function D_d_Featurs = Delta_DeltaFeature (delta_window, Featurs) % Compute delta
                                                                  delta cepstrum and
                                                                  delta log energy.

   rows = size (Featurs, 1);
   cols  = size(Featurs,2);
   temp1  = [zeros(rows,delta_window) Featurs zeros(rows,delta_window)];
   temp2  = [zeros(rows,delta_window) Featurs zeros(rows,delta_window)];
   D_d_Featurs  = zeros(rows,cols);

   % Rabiner method

   denominator = sum ([1:delta_window].^2)*2;
   denominator2 =
delta_window*(delta_window+1)*(2*delta_window+1)*(3*delta_window^2+3*delta
_window-1)/15;
   for i = 1+delta_window : cols+delta_window,
     subtrahend = 0;
     minuend    = 0;
     subtrahend2 = 0;
     minuend2    = 0;
     for j = 1 : delta_window,
       subtrahend = subtrahend + temp1(:,i+j);
       minuend = minuend + temp1(:,i-j);
       subtrahend2  =  subtrahend2+ j*j*temp2(:,i+j);
       minuend2    = minuend2 + (-j)*(-j)*temp2(:,i-j);
     end;
     temp1(:,i) = subtrahend + minuend + temp1(:,i);
```

```
      temp2(:,i) = subtrahend2 + minuend2;
      D_d_Featurs(:,i-delta_window) = 2*(denominator.*temp1(:,i)-
(2*delta_window+1).*temp2(:,i))/(denominator*denominator-
(2*delta_window+1)*denominator2);
   end;
```

7) Main_DTW.m

```
function Main_DTW(Option)
No_Templates=60;                          % total number of references template
 switch Option
    case {'Record_mode'}
       Fs=16000;
       Duration=2; % in seconds
       TestWave=Record (Fs, Duration); % record a wave file in Fs Hz, with Duration
    case {'Open_mode'}
        [TestWave, Fs]=OpenWave;   % Open a wanted wave file
     otherwise
         disp('Default: 1_ChangeChannel.wav');
         TestWaveName='1_ChangeChannel.wav';
         [TestWave,Fs,NBits] = wavread(TestWaveName);
         wavplay(TestWave, Fs, 'sync');

         % ===== Plot recorded waveform
         hold on
         plot (TestWave,'b--');
         title ('Recorded wave file');
         axis ([0 Duration min (Out) max(Out)]);
         grid on
         hold off

         % ==== Save recorded file
         FileName = 'Test.wav';
         wavwrite(TestWave, Fs, 60, FileName);
         fprintf('    >> The file is save to "%s"\n', FileName);
 end

disp(
'=======================================================');
disp('  Start recognizing by DTW (39 CMS-MFCC features)... ');
Test_MFCC_Features= CMS_Normalization(Feature_Extruction(TestWave,Fs));

for i=1:No_Templates
      [Template_MFCC_Features,Template_Name]=SelectNextTemplate(i);
    % Construct the 'local match' scores matrix as the cosine distance between the features
      Local_Distance =
LocalDistance(abs(Template_MFCC_Features),abs(Test_MFCC_Features));
```

```
    % Find the lowest-cost path across Local_Distance matrix
    [Path_y,Path_x,Distance] = DTW(Local_Distance);
    % Least cost (final cost) is value in top right corner of Distance matrix
    Distance_from_Template(i)=Distance(1,size(Distance,2));
    if i>1
       if Distance_from_Template(i)<Answer_DistanceFrom
          Answer_Name=Template_Name;
          Answer_Distance=Distance;
          Answer_Path_x=Path_x;
          Answer_Path_y=Path_y;
          Answer_DistanceFrom=Distance_from_Template(i);
       end
    else
       Answer_Name=Template_Name;
       Answer_Distance=Distance;
       Answer_Path_x=Path_x;
       Answer_Path_y=Path_y;
       Answer_DistanceFrom=Distance_from_Template(i);
    end

    % Plot the min cost path though Distance matrix for all Templates
    colormap(1-gray);
    subplot(3,3,i);
    imagesc(Distance)
    hold on; plot(Path_x,Path_y,'r'); hold off
    str=['DTW for ',Template_Name,' ,Distance =
',num2str(Distance_from_Template(i))];
    title(Template_Name);
  end

  % Plot the min cost path though Distance matrix for Answer (Template with MIN final cost)

 figure(3);
  colormap('jet');
  imagesc(Answer_Distance)
  hold on; plot(Answer_Path_x,Answer_Path_y,'r'); hold off
  str=['Answer is: ',Answer_Name,' ,Distance= ',num2str(Answer_DistanceFrom)];
  title(str);
  disp([' It''s seem that answer is: <<',Answer_Name,'>>, Am I right :=# ?']);




% clear dummy variables

   clear TestWave Fs  Duration Template_List No_Templates i str Path_y Path_x
Template_MFCC_Features Template_Name Local_Distance Distance
Test_MFCC_Features %Distance_from_Template
```

```
   clear  Template_MFCC_Features_1_Buka Template_MFCC_Features_1_Tutup
Template_MFCC_Features_1_KuatkanSuara
Template_MFCC_Features_1_PerlahankanSuara
Template_MFCC_Features_1_TukarSiaran
   clear  Template_MFCC_Features_2_Buka Template_MFCC_Features_2_Tutup
Template_MFCC_Features_2_KuatkanSuara
Template_MFCC_Features_2_PerlahankanSuara
Template_MFCC_Features_2_TukarSiaran
   clear  Template_MFCC_Features_3_Buka Template_MFCC_Features_3_Tutup
Template_MFCC_Features_3_KuatkanSuara
Template_MFCC_Features_3_PerlahankanSuara
Template_MFCC_Features_3_TukarSiaran
   clear  Template_MFCC_Features_4_Buka Template_MFCC_Features_4_Tutup
Template_MFCC_Features_4_KuatkanSuara
Template_MFCC_Features_4_PerlahankanSuara
Template_MFCC_Features_4_TukarSiaran
  clear  Template_MFCC_Features_5_Buka Template_MFCC_Features_5_Tutup
Template_MFCC_Features_5_KuatkanSuara
Template_MFCC_Features_5_PerlahankanSuara
Template_MFCC_Features_5_TukarSiaran
  clear  Template_MFCC_Features_6_Buka Template_MFCC_Features_6_Tutup
Template_MFCC_Features_6_KuatkanSuara
Template_MFCC_Features_6_PerlahankanSuara
Template_MFCC_Features_6_TukarSiaran
   clear  Template_MFCC_Features_7_Buka Template_MFCC_Features_7_Tutup
Template_MFCC_Features_7_KuatkanSuara
Template_MFCC_Features_7_PerlahankanSuara
Template_MFCC_Features_7_TukarSiaran
   clear  Template_MFCC_Features_8_Buka Template_MFCC_Features_8_Tutup
Template_MFCC_Features_8_KuatkanSuara
Template_MFCC_Features_8_PerlahankanSuara
Template_MFCC_Features_8_TukarSiaran
   clear  Template_MFCC_Features_9_Buka Template_MFCC_Features_9_Tutup
Template_MFCC_Features_9_KuatkanSuara
Template_MFCC_Features_9_PerlahankanSuara
Template_MFCC_Features_9_TukarSiaran
  clear  Template_MFCC_Features_10_Buka
Template_MFCC_Features_10_TutupTemplate_MFCC_Features_10_KuatkanSuara
Template_MFCC_Features_10_PerlahankanSuara
Template_MFCC_Features_10_TukarSiaran
  clear  Template_MFCC_Features_11_Buka Template_MFCC_Features_11_Tutup
Template_MFCC_Features_11_KuatkanSuara
Template_MFCC_Features_11_PerlahankanSuara
Template_MFCC_Features_11_TukarSiaran
  clear  Template_MFCC_Features_12_Buka Template_MFCC_Features_12_Tutup
Template_MFCC_Features_12_KuatkanSuara
Template_MFCC_Features_12_PerlahankanSuara
Template_MFCC_Features_12_TukarSiaran
```

8) DTW.m

```
function [Path_y,Path_x,Distance] = DTW(LocalDistance)
   % [Path_y,Path_x] = DTW(LocalDistance)
   %    Use dynamic programming to find a min-cost path through matrix Local Distance.
   %    Return state sequence in Path_y,Path_x

   [Row,Col] = size(LocalDistance);
      Distance = zeros(Row+1, Col+1);
   Distance(Row+1,:) = NaN;
   Distance(:,1) = NaN;
   Distance(Row+1,1) = 0;
   Distance(1:(Row), 2:(Col+1)) = LocalDistance;
   AllPath = zeros(Row,Col);
   for i = Row+1:-1:2;
     for j = 1:Col;
       [SelPath, tb] = min([Distance(i, j), Distance(i, j+1), Distance(i-1, j)]);
       Distance(i-1,j+1) = Distance(i-1,j+1)+SelPath;
       AllPath(i-1,j) = tb;
     end
   end

   % Traceback from top left for finding Path

   i = 1;
   j = Col;
   Path_y = i;
   Path_x = j;
   while i < Row & j > 1
     tb = AllPath(i,j);
     if (tb == 1)
       i = i+1;
       j = j-1;
     elseif (tb == 2)
       i = i+1;
     elseif (tb == 3)
       j = j-1;
     else
       error;
     end
     Path_y = [i,Path_y];
     Path_x = [j,Path_x];
   end

   Distance = Distance(1:(Row),2:(Col+1));
```

9) Buffer2.m

```
function [Path_y,Path_x,Distance] = DTW(LocalDistance)

   % [Path_y,Path_x] = DTW(LocalDistance)
```

```
%   Use dynamic programming to find a min-cost path through matrix Local Distance.
%   Return state sequence in Path_y,Path_x

[Row,Col] = size(LocalDistance);

% costs
Distance = zeros(Row+1, Col+1);
Distance(Row+1,:) = NaN;
Distance(:,1) = NaN;
Distance(Row+1,1) = 0;
Distance(1:(Row), 2:(Col+1)) = LocalDistance;

AllPath = zeros(Row,Col);

for i = Row+1:-1:2;
  for j = 1:Col;
    [SelPath, tb] = min([Distance(i, j), Distance(i, j+1), Distance(i-1, j)]);
    Distance(i-1,j+1) = Distance(i-1,j+1)+SelPath;
    AllPath(i-1,j) = tb;
  end
end

% Traceback from top left for finding Path

i = 1;
j = Col;
Path_y = i;
Path_x = j;
while i < Row & j > 1
  tb = AllPath(i,j);
  if (tb == 1)
    i = i+1;
    j = j-1;
  elseif (tb == 2)
    i = i+1;
  elseif (tb == 3)
    j = j-1;
  else
    error;
  end
  Path_y = [i,Path_y];
  Path_x = [j,Path_x];
end

Distance = Distance(1:(Row),2:(Col+1));
```

10) CMS_Normalization.m

```
function Out=CMS_Normalization(Featurs) % Cepstral Mean Subtraction (CMS) of
                                        Features

   [N,M]=size(Featurs);

   for i=1:N
      Mean(i)=sum(Featurs(i,:))/M;
      Out(i,:)=Featurs(i,:)-Mean(i);
   end
```

11) LocalDistance.m

```
function Out2 = LocalDistance(A,B)

 % Out = LocalDistance(A,B)
 %  calculates the local distance between feature matrices A and B.
 %  using inner product i.e. cos(angle between vectors) between vectors.
 %   A and B have same rows.
   Mag_A = sqrt(sum(A.^2));
   Mag_B = sqrt(sum(B.^2));
   Cols_A = size(A,2);
   Cols_B = size(B,2);
   Out = zeros(Cols_A, Cols_B);
   for i = 1:Cols_A
    for j = 1:Cols_B

      % normalized inner product i.e. cos(angle between vectors)

      Out(i,j) = (A(:,i)'*B(:,j))/(Mag_A(i)*Mag_B(j));
    end
   end

   Row=size(Out,1);
   for i=1:fix(Row/2)
      Out2(Row-i+1,:)=Out(i,:);
      Out2(i,:)=Out(Row-i+1,:);%tmp;
   end
   if mod(Row,2)~=0
      Out2(fix(Row/2+1),:)=Out(fix(Row/2+1),:);
   end

  % Use 1-Out2 because DTW will find the *lowest* total cost
   Out2=1-Out2;
```

APPENDIX C: MATLAB CODE FOR VQ

This appendix provides a matlab code for VQ technique.

1) Train.m

```
function code = train(traindir, n)
% Input:
% traindir : string name of directory contains all train sound files
%  n       : number of train files in traindir
%
% Output:
%       code     : trained VQ codebooks, code{i} for i-th speaker
%
% Note:
%       Sound files in traindir is supposed to be:
%                   s1.wav, s2.wav, ..., sn.wav
% Example:
%       >> code = train('D:\train\', 8);
%%%%%%%%%%%%%%%%%%%%

k = 16;                     % number of centroids required
for i = 1:n                 % train a VQ codebook for each speaker
   file = sprintf('%ss%d.wav', traindir, i);
   disp(file);
  s  = wavread(file);
 fs=8000;
v= Feature_Extruction(s,fs);                              % MFCC only
v= CMS_Normalization(Feature_Extruction(s,fs));   % MFCCCMS
   code{i} = vqlbg(v, k);                                 % Train VQ codebook
end
```

2) Test.m

```
function test(testdir, n, code)

%===================================================
% Input:
%    testdir : string name of directory contains all test sound files
%       n       : number of test files in testdir
%       code    : codebooks of all trained speakers
% Note:
%       Sound files in testdir is supposed to be:
%              s1.wav, s2.wav, ..., sn.wav
%
% Example:
%       >> test('D:\train\', 8, code);
%===================================================
```

```
for k = 1:n                    % read test sound file of each speaker
   file = sprintf('%ss%d.wav', testdir, k);

 %  [s fs] = wavread(file);
  s  = wavread(file);
 fs=8000;
v=Feature_Extruction(s,fs);
 %v= CMS_Normalization(Feature_Extruction(s,fs));
   distmin = inf;
   k1 = 0;
   for l = 1:length(code) % each trained codebook,compute distortion
      d = disteu(v, code{l});
      dist = sum(min(d,[],2)) / size(d,1);

      if dist < distmin
         distmin = dist;
         k1 = l;
      end
   end

   msg = sprintf('word %d matches with word %d', k, k1);
   disp(msg);
end
```

3) blockFrames.m

```
function M3 = blockFrames(s, fs, m, n)
%=================================================
% blockFrames: Puts the signal into frames
% Inputs: s  contains the signal to analize
%         fs is the sampling rate of the signal
%         m  is the distance between the beginnings of two frames
%         n  is the number of samples per frame
% Output: M3 is a matrix containing all the frames
%=================================================

   l = length(s);
   nbFrame = floor((l - n) / m) + 1;

   for i = 1:n
      for j = 1:nbFrame
         M(i, j) = s(((j - 1) * m) + i);
      end
   end

   h = hamming(n);
   M2 = diag(h) * M;
```

```
   for i = 1:nbFrame
      M3(:, i) = fft(M2(:, i));
   end
```

4) Buffer2.m

```
function out = buffer2(y, frameSize, overlap)

 % buffer2: Frame blocking
%    Usage: out = buffer2(y, frameSize, overlap)

if nargin<3, overlap=0; end
 if nargin<2, frameSize=256; end
 y = y(:);
 step = frameSize-overlap;
frameCount = floor((length(y)-overlap)/step);

 out = zeros(frameSize, frameCount);
 for i=1:frameCount,
   startIndex = (i-1)*step+1;
   out(:, i) = y(startIndex:(startIndex+frameSize-1));
 end
```

5) CMS_Normalization.m

```
function Out=CMS_Normalization(Featurs)
   % Cepstral Mean Subtraction (CMS) of Features
   [N,M]=size(Featurs);
   for i=1:N
      Mean(i)=sum(Featurs(i,:))/M;
      Out(i,:)=Featurs(i,:)-Mean(i);
   end
```

6) disteu.m

```
function d = disteu(x, y)
% DISTEU Pairwise Euclidean distances between columns of two matrices
%
% Input: x, y:   Two matrices whose each column is an a vector data.
%
% Output,d =Element d(i,j) will be the Euclidean distance between two
%         column vectors X(:,i) and Y(:,j)
%
% Note:
%       The Euclidean distance D between two vectors X and Y is:
%       D = sum((x-y).^2).^0.5
%%%%%%%%%%%%%%%%%%%%

[M, N] = size(x);
[M2, P] = size(y);
```

```
if (M ~= M2)
   error('Matrix dimensions do not match.')
end

d = zeros(N, P);

if (N < P)
   copies = zeros(1,P);
   for n = 1:N
      d(n,:) = sum((x(:, n+copies) - y) .^2, 1);
   end
else
   copies = zeros(1,N);
   for p = 1:P
      d(:,p) = sum((x - y(:, p+copies)) .^2, 1)';
   end
end

d = d.^0.5;
```

APPENDIX D: DTW TEMPLATE FOR MALAY AND ENGLISH WORDS

This appendix provides template for English and Malay words using DTW technique.

1) English word

```
% Dynamic Time Warping (DTW)
 % Extructing features of Templates and save them.

 clear all;
 close all;
clc;
 %Path='D:\MYProject\Voice_DTW-Versi-Linda\Data\';
 Path='Data\';
Template_List=['1_SwitchOn1','1_SwitchOn2','1_SwitchOn3','1_SwitchOn4','1_Swit
chOff1','1_SwitchOff2','1_SwitchOff3','1_SwitchOff4','1_VolumeUp1','1_VolumeUp
2','1_VolumeUp3','1_VolumeUp4','1_VolumeDown1','1_VolumeDown2','1_Volume
Down3','1_VolumeDown4','1_ChangeChannel','1_ChangeChannel2','1_ChangeChann
el3','1_ChangeChannel4','2_SwitchOn1','2_SwitchOn2','2_SwitchOn3','2_SwitchOn4',
'2_SwitchOff1','2_SwitchOff2','2_SwitchOff3','2_SwitchOff4','2_VolumeUp1','2_Vol
umeUp2','2_VolumeUp3','2_VolumeUp4','2_VolumeDown1','2_VolumeDown2','2_V
olumeDown3','2_VolumeDown4','2_ChangeChannel','2_ChangeChannel2','2_Change
Channel3','2_ChangeChannel4','3_SwitchOn1','3_SwitchOn2','3_SwitchOn3','3_Switc
hOn4','3_SwitchOff1','3_SwitchOff2','3_SwitchOff3','3_SwitchOff4','3_VolumeUp1','
3_VolumeUp2','3_VolumeUp3','3_VolumeUp4','3_VolumeDown1','3_VolumeDown2
','3_VolumeDown3','3_VolumeDown4','3_ChangeChannel','3_ChangeChannel2','3_C
hangeChannel3','3_ChangeChannel4','4_SwitchOn1','4_SwitchOn2','4_SwitchOn3','4
_SwitchOn4','4_SwitchOff1','4_SwitchOff2','4_SwitchOff3','4_SwitchOff4','4_Volum
eUp1','4_VolumeUp2','4_VolumeUp3','4_VolumeUp4','4_VolumeDown1','4_Volume
Down2','4_VolumeDown3','4_VolumeDown4','4_ChangeChannel','4_ChangeChannel
2','4_ChangeChannel3','4_ChangeChannel4','5_SwitchOn1','5_SwitchOn2','5_Switch
On3','5_SwitchOn4','5_SwitchOff1','5_SwitchOff2','5_SwitchOff3','5_SwitchOff4','5_
VolumeUp1','5_VolumeUp2','5_VolumeUp3','5_VolumeUp4','5_VolumeDown1','5_
VolumeDown2','5_VolumeDown3','5_VolumeDown4','5_ChangeChannel','5_Change
Channel2','5_ChangeChannel3','5_ChangeChannel4','6_SwitchOn1','6_SwitchOn2','6
_SwitchOn3','6_SwitchOn4','6_SwitchOff1','6_SwitchOff2','6_SwitchOff3','6_Switch
Off4','6_VolumeUp1','6_VolumeUp2','6_VolumeUp3','6_VolumeUp4','6_VolumeDo
wn1','6_VolumeDown2','6_VolumeDown3','6_VolumeDown4','6_ChangeChannel','6
_ChangeChannel2','6_ChangeChannel3','6_ChangeChannel4','7_SwitchOn1','7_Switc
hOn2','7_SwitchOn3','7_SwitchOn4','7_SwitchOff1','7_SwitchOff2','7_SwitchOff3','7
_SwitchOff4','7_VolumeUp1','7_VolumeUp2','7_VolumeUp3','7_VolumeUp4','7_Vol
umeDown1','7_VolumeDown2','7_VolumeDown3','7_VolumeDown4','7_ChangeCha
nnel','7_ChangeChannel2','7_ChangeChannel3','7_ChangeChannel4','8_SwitchOn1','8
_SwitchOn2','8_SwitchOn3','8_SwitchOn4','8_SwitchOff1','8_SwitchOff2','8_Switch
Off3','8_SwitchOff4','8_VolumeUp1','8_VolumeUp2','8_VolumeUp3','8_VolumeUp4'
,'8_VolumeDown1','8_VolumeDown2','8_VolumeDown3','8_VolumeDown4','8_Cha
ngeChannel','8_ChangeChannel2','8_ChangeChannel3','8_ChangeChannel4','9_Switc
```

```
hOn1','9_SwitchOn2','9_SwitchOn3','9_SwitchOn4','9_SwitchOff1','9_SwitchOff2','9
_SwitchOff3','9_SwitchOff4','9_VolumeUp1','9_VolumeUp2','9_VolumeUp3','9_Vol
umeUp4','9_VolumeDown1','9_VolumeDown2','9_VolumeDown3','9_VolumeDown4
','9_ChangeChannel','9_ChangeChannel2','9_ChangeChannel3','9_ChangeChannel4','1
0_SwitchOn1','10_SwitchOn2','10_SwitchOn3','10_SwitchOn4','10_SwitchOff1','10_
SwitchOff2','10_SwitchOff3','10_SwitchOff4','10_VolumeUp1','10_VolumeUp2','10_
VolumeUp3','10_VolumeUp4','10_VolumeDown1','10_VolumeDown2','10_VolumeD
own3','10_VolumeDown4','10_ChangeChannel','10_ChangeChannel2','10_ChangeCh
annel3','10_ChangeChannel4','11_SwitchOn1','11_SwitchOn2','11_SwitchOn3','11_S
witchOn4','11_SwitchOff1','11_SwitchOff2','11_SwitchOff3','11_SwitchOff4','11_Vo
lumeUp1','11_VolumeUp2','11_VolumeUp3','11_VolumeUp4','11_VolumeDown1','1
1_VolumeDown2','11_VolumeDown3','11_VolumeDown4','11_ChangeChannel','11_
ChangeChannel2','11_ChangeChannel3','11_ChangeChannel4'];

 Template_MFCC_Features_zero=Feature_Extruction([Path,'1_SwitchOn1.wav']);
 Template_MFCC_Features_one= Feature_Extruction([Path,'1_SwitchOn2.wav']);
 Template_MFCC_Features_two=Feature_Extruction([Path,'1_SwitchOn3.wav']);
 Template_MFCC_Features_three= Feature_Extruction([Path,'1_SwitchOn4.wav']);
 Template_MFCC_Features_four= Feature_Extruction([Path,'1_SwitchOff1.wav']);
 Template_MFCC_Features_five= Feature_Extruction([Path,'1_SwitchOff2.wav']);
 Template_MFCC_Features_six= Feature_Extruction([Path,'1_SwitchOff3.wav']);
 Template_MFCC_Features_seven= Feature_Extruction([Path,'1_SwitchOff4.wav']);
 Template_MFCC_Features_eight= Feature_Extruction([Path,'1_VolumeUp1.wav']);
 Template_MFCC_Features_nine= Feature_Extruction([Path,'1_VolumeUp2.wav']);
 Template_MFCC_Features_ten=Feature_Extruction([Path,'1_VolumeUp3.wav']);
 Template_MFCC_Features_oneone=
Feature_Extruction([Path,'1_VolumeUp4.wav']);
 Template_MFCC_Features_onetwo=
Feature_Extruction([Path,'1_VolumeDown1.wav']);

Template_MFCC_Features_onethree=Feature_Extruction([Path,'1_VolumeDown2.w
av']);
 Template_MFCC_Features_onefour=
Feature_Extruction([Path,'1_VolumeDown3.wav']);
 Template_MFCC_Features_onefive=
Feature_Extruction([Path,'1_VolumeDown4.wav']);
 Template_MFCC_Features_onesix=
Feature_Extruction([Path,'1_ChangeChannel1.wav']);
 Template_MFCC_Features_oneseven=
Feature_Extruction([Path,'1_ChangeChannel2.wav']);
 Template_MFCC_Features_oneeight=
Feature_Extruction([Path,'1_ChangeChannel3.wav']);

Template_MFCC_Features_onenine=Feature_Extruction([Path,'1_ChangeChannel4.
wav']);

 Template_MFCC_Features_twoone= Feature_Extruction([Path,'2_SwitchOn1.wav']);
 Template_MFCC_Features_twotwo=
Feature_Extruction([Path,'2_SwitchOn2.wav']);
```

```
 Template_MFCC_Features_twothree=
Feature_Extruction([Path,'2_SwitchOn3.wav']);
 Template_MFCC_Features_twofour=
Feature_Extruction([Path,'2_SwitchOn4.wav']);
 Template_MFCC_Features_twofive=
Feature_Extruction([Path,'2_SwitchOff1.wav']);
 Template_MFCC_Features_twosix= Feature_Extruction([Path,'2_SwitchOff2.wav']);
 Template_MFCC_Features_twoseven=
Feature_Extruction([Path,'2_SwitchOff3.wav']);
 Template_MFCC_Features_twoeight=
Feature_Extruction([Path,'2_SwitchOff4.wav']);
 Template_MFCC_Features_twonine=
Feature_Extruction([Path,'2_VolumeUp1.wav']);
 Template_MFCC_Features_threeone=
Feature_Extruction([Path,'2_VolumeUp2.wav']);
 Template_MFCC_Features_threetwo=
Feature_Extruction([Path,'2_VolumeUp3.wav']);
 Template_MFCC_Features_threethree=
Feature_Extruction([Path,'2_VolumeUp4.wav']);
 Template_MFCC_Features_threefour=
Feature_Extruction([Path,'2_VolumeDown1.wav']);
 Template_MFCC_Features_threefive=
Feature_Extruction([Path,'2_VolumeDown2.wav']);
 Template_MFCC_Features_threesix=
Feature_Extruction([Path,'2_VolumeDown3.wav']);
 Template_MFCC_Features_threeseven=
Feature_Extruction([Path,'2_VolumeDown4.wav']);
 Template_MFCC_Features_threeeight=
Feature_Extruction([Path,'2_ChangeChannel1.wav']);
 Template_MFCC_Features_threenine=
Feature_Extruction([Path,'2_ChangeChannel2.wav']);
 Template_MFCC_Features_fourone=
Feature_Extruction([Path,'2_ChangeChannel3.wav']);
 Template_MFCC_Features_fourtwo=
Feature_Extruction([Path,'2_ChangeChannel4.wav']);

 Template_MFCC_Features_fourthree=
Feature_Extruction([Path,'3_SwitchOn1.wav']);
 Template_MFCC_Features_fourfour=
Feature_Extruction([Path,'3_SwitchOn2.wav']);
 Template_MFCC_Features_fourfive=
Feature_Extruction([Path,'3_SwitchOn3.wav']);
 Template_MFCC_Features_foursix= Feature_Extruction([Path,'3_SwitchOn4.wav']);
 Template_MFCC_Features_fourseven=
Feature_Extruction([Path,'3_SwitchOff1.wav']);
 Template_MFCC_Features_foureight=
Feature_Extruction([Path,'3_SwitchOff2.wav']);
 Template_MFCC_Features_fournine=
Feature_Extruction([Path,'3_SwitchOff3.wav']);
```

```
 Template_MFCC_Features_fiveone=
Feature_Extruction([Path,'3_SwitchOff4.wav']);
 Template_MFCC_Features_fivetwo=
Feature_Extruction([Path,'3_VolumeUp1.wav']);
 Template_MFCC_Features_fivethree=
Feature_Extruction([Path,'3_VolumeUp2.wav']);
 Template_MFCC_Features_fivefour=
Feature_Extruction([Path,'3_VolumeUp3.wav']);
 Template_MFCC_Features_fivefive=
Feature_Extruction([Path,'3_VolumeUp4.wav']);
 Template_MFCC_Features_fivesix=
Feature_Extruction([Path,'3_VolumeDown1.wav']);
 Template_MFCC_Features_fiveseven=
Feature_Extruction([Path,'3_VolumeDown2.wav']);
 Template_MFCC_Features_fiveeight=
Feature_Extruction([Path,'3_VolumeDown3.wav']);
 Template_MFCC_Features_fivenine=
Feature_Extruction([Path,'3_VolumeDown4.wav']);
 Template_MFCC_Features_sixone=
Feature_Extruction([Path,'3_ChangeChannel1.wav']);
 Template_MFCC_Features_sixtwo=
Feature_Extruction([Path,'3_ChangeChannel2.wav']);
 Template_MFCC_Features_sixthree=
Feature_Extruction([Path,'3_ChangeChannel3.wav']);
 Template_MFCC_Features_sixfour=
Feature_Extruction([Path,'3_ChangeChannel4.wav']);

 Template_MFCC_Features_sixfive= Feature_Extruction([Path,'4_SwitchOn1.wav']);
 Template_MFCC_Features_sixsix= Feature_Extruction([Path,'4_SwitchOn2.wav']);
 Template_MFCC_Features_sixseven=
Feature_Extruction([Path,'4_SwitchOn3.wav']);
 Template_MFCC_Features_sixeight=
Feature_Extruction([Path,'4_SwitchOn4.wav']);
 Template_MFCC_Features_sixnine=
Feature_Extruction([Path,'4_SwitchOff1.wav']);
 Template_MFCC_Features_sevenone=
Feature_Extruction([Path,'4_SwitchOff2.wav']);
 Template_MFCC_Features_seventwo=
Feature_Extruction([Path,'4_SwitchOff3.wav']);
 Template_MFCC_Features_seventhree=
Feature_Extruction([Path,'4_SwitchOff4.wav']);
 Template_MFCC_Features_sevenfour=
Feature_Extruction([Path,'4_VolumeUp1.wav']);
 Template_MFCC_Features_sevenfive=
Feature_Extruction([Path,'4_VolumeUp2.wav']);
 Template_MFCC_Features_sevensix=
Feature_Extruction([Path,'4_VolumeUp3.wav']);
 Template_MFCC_Features_sevenseven=
Feature_Extruction([Path,'4_VolumeUp4.wav']);
```

```
 Template_MFCC_Features_seveneight
=Feature_Extruction([Path,'4_VolumeDown1.wav']);
 Template_MFCC_Features_sevennine=
Feature_Extruction([Path,'4_VolumeDown2.wav']);
 Template_MFCC_Features_eightone=
Feature_Extruction([Path,'4_VolumeDown3.wav']);
 Template_MFCC_Features_eighttwo=
Feature_Extruction([Path,'4_VolumeDown4.wav']);
 Template_MFCC_Features_eightthree=
Feature_Extruction([Path,'4_ChangeChannel1.wav']);
 Template_MFCC_Features_eightfour=
Feature_Extruction([Path,'4_ChangeChannel2.wav']);
 Template_MFCC_Features_eightfive=
Feature_Extruction([Path,'4_ChangeChannel3.wav']);
 Template_MFCC_Features_eightsix=
Feature_Extruction([Path,'4_ChangeChannel4.wav']);

 Template_MFCC_Features_eightseven=
Feature_Extruction([Path,'5_SwitchOn1.wav']);
 Template_MFCC_Features_eighteight=
Feature_Extruction([Path,'5_SwitchOn2.wav']);
 Template_MFCC_Features_eightnine=
Feature_Extruction([Path,'5_SwitchOn3.wav']);
 Template_MFCC_Features_nineone=
Feature_Extruction([Path,'5_SwitchOn4.wav']);
 Template_MFCC_Features_ninetwo=
Feature_Extruction([Path,'5_SwitchOff1.wav']);
 Template_MFCC_Features_ninethree=
Feature_Extruction([Path,'5_SwitchOff2.wav']);
 Template_MFCC_Features_ninefour=
Feature_Extruction([Path,'5_SwitchOff3.wav']);
 Template_MFCC_Features_ninefive=
Feature_Extruction([Path,'5_SwitchOff4.wav']);
 Template_MFCC_Features_ninesix=
Feature_Extruction([Path,'5_VolumeUp1.wav']);
 Template_MFCC_Features_nineseven=
Feature_Extruction([Path,'5_VolumeUp2.wav']);
 Template_MFCC_Features_nineeight=
Feature_Extruction([Path,'5_VolumeUp3.wav']);
 Template_MFCC_Features_ninenine=
Feature_Extruction([Path,'5_VolumeUp4.wav']);
 Template_MFCC_Features_tenone=
Feature_Extruction([Path,'5_VolumeDown1.wav']);
 Template_MFCC_Features_tentwo=
Feature_Extruction([Path,'5_VolumeUp2.wav']);
 Template_MFCC_Features_tenthree=
Feature_Extruction([Path,'5_VolumeUp3.wav']);
 Template_MFCC_Features_tenfour=
Feature_Extruction([Path,'5_VolumeUp4.wav']);
```

```
 Template_MFCC_Features_tenfive=
Feature_Extruction([Path,'5_ChangeChannel1.wav']);
 Template_MFCC_Features_tensix=
Feature_Extruction([Path,'5_ChangeChannel2.wav']);
 Template_MFCC_Features_tenseven=
Feature_Extruction([Path,'5_ChangeChannel3.wav']);
 Template_MFCC_Features_teneight=
Feature_Extruction([Path,'5_ChangeChannel4.wav']);

  Template_MFCC_Features_tennine=
Feature_Extruction([Path,'6_SwitchOn1.wav']);
 Template_MFCC_Features_elevenone=
Feature_Extruction([Path,'6_SwitchOn2.wav']);
 Template_MFCC_Features_eleventwo=
Feature_Extruction([Path,'6_SwitchOn3.wav']);
 Template_MFCC_Features_eleventhree=
Feature_Extruction([Path,'6_SwitchOn4.wav']);
 Template_MFCC_Features_elevenfour=
Feature_Extruction([Path,'6_SwitchOff1.wav']);
 Template_MFCC_Features_elevenfive=
Feature_Extruction([Path,'6_SwitchOff2.wav']);
 Template_MFCC_Features_elevensix=
Feature_Extruction([Path,'6_SwitchOff3.wav']);
 Template_MFCC_Features_elevenseven=
Feature_Extruction([Path,'6_SwitchOff4.wav']);
 Template_MFCC_Features_eleveneight=
Feature_Extruction([Path,'6_VolumeUp1.wav']);
 Template_MFCC_Features_elevennine=
Feature_Extruction([Path,'6_VolumeUp2.wav']);
 Template_MFCC_Features_twelveone=
Feature_Extruction([Path,'6_VolumeUp3.wav']);
 Template_MFCC_Features_twelvetwo=
Feature_Extruction([Path,'6_VolumeUp4.wav']);
 Template_MFCC_Features_twelvethree=
Feature_Extruction([Path,'6_VolumeDown1.wav']);
 Template_MFCC_Features_twelvefour=
Feature_Extruction([Path,'6_VolumeDown2.wav']);
 Template_MFCC_Features_twelvefive=
Feature_Extruction([Path,'6_VolumeDown3.wav']);
 Template_MFCC_Features_twelvesix=
Feature_Extruction([Path,'6_VolumeDown4.wav']);
 Template_MFCC_Features_twelveseven=
Feature_Extruction([Path,'6_ChangeChannel1.wav']);
 Template_MFCC_Features_twelveeight=
Feature_Extruction([Path,'6_ChangeChannel2.wav']);
 Template_MFCC_Features_twelvenine=
Feature_Extruction([Path,'6_ChangeChannel3.wav']);
 Template_MFCC_Features_thirteenone=
Feature_Extruction([Path,'6_ChangeChannel4.wav']);
```

```
 Template_MFCC_Features_thirteentwo=
Feature_Extruction([Path,'7_SwitchOn1.wav']);
 Template_MFCC_Features_thirteenthree=
Feature_Extruction([Path,'7_SwitchOn2.wav']);
 Template_MFCC_Features_thirteenfour=
Feature_Extruction([Path,'7_SwitchOn3.wav']);
 Template_MFCC_Features_thirteenfive=
Feature_Extruction([Path,'7_SwitchOn4.wav']);
 Template_MFCC_Features_thirteensix=
Feature_Extruction([Path,'7_SwitchOff1.wav']);
 Template_MFCC_Features_thirteenseven=
Feature_Extruction([Path,'7_SwitchOff2.wav']);
 Template_MFCC_Features_thirteeneight=
Feature_Extruction([Path,'7_SwitchOff3.wav']);
 Template_MFCC_Features_thirteennine=
Feature_Extruction([Path,'7_SwitchOff4.wav']);
 Template_MFCC_Features_fourteenone=
Feature_Extruction([Path,'7_VolumeUp1.wav']);
 Template_MFCC_Features_fourteentwo=
Feature_Extruction([Path,'7_VolumeUp2.wav']);
 Template_MFCC_Features_fourteenthree=
Feature_Extruction([Path,'7_VolumeUp3.wav']);
 Template_MFCC_Features_fourteenfour                                    =
Feature_Extruction([Path,'7_VolumeUp4.wav']);
 Template_MFCC_Features_fourteenfive=
Feature_Extruction([Path,'7_VolumeDown1.wav']);
 Template_MFCC_Features_fourteensix=
Feature_Extruction([Path,'7_VolumeDown2.wav']);
 Template_MFCC_Features_fourteenseven=
Feature_Extruction([Path,'7_VolumeDown3.wav']);
 Template_MFCC_Features_fourteeneight=
Feature_Extruction([Path,'7_VolumeDown4.wav']);
 Template_MFCC_Features_fourteennine=
Feature_Extruction([Path,'7_ChangeChannel1.wav']);
 Template_MFCC_Features_fifthteenone=
Feature_Extruction([Path,'7_ChangeChannel2.wav']);
 Template_MFCC_Features_fifthteentwo=
Feature_Extruction([Path,'7_ChangeChannel3.wav']);
 Template_MFCC_Features_fifthteenthree=
Feature_Extruction([Path,'7_ChangeChannel4.wav']);
 Template_MFCC_Features_fifthteenfour=
Feature_Extruction([Path,'8_SwitchOn1.wav']);
 Template_MFCC_Features_fifthteenfive=
Feature_Extruction([Path,'8_SwitchOn2.wav']);
 Template_MFCC_Features_fifthteensix=
Feature_Extruction([Path,'8_SwitchOn3.wav']);
 Template_MFCC_Features_fifthteenseven=
Feature_Extruction([Path,'8_SwitchOn4.wav']);
```

```
 Template_MFCC_Features_fifthteeneight=
Feature_Extruction([Path,'8_SwitchOff1.wav']);
 Template_MFCC_Features_fifthteennine=
Feature_Extruction([Path,'8_SwitchOff2.wav']);
 Template_MFCC_Features_sixteenone=
Feature_Extruction([Path,'8_SwitchOff3.wav']);
 Template_MFCC_Features_sixteentwo=
Feature_Extruction([Path,'8_SwitchOff4.wav']);
 Template_MFCC_Features_sixteenthree=
Feature_Extruction([Path,'8_VolumeUp1.wav']);
 Template_MFCC_Features_sixteenfour=
Feature_Extruction([Path,'8_VolumeUp2.wav']);
 Template_MFCC_Features_sixteenfive=
Feature_Extruction([Path,'8_VolumeUp3.wav']);
 Template_MFCC_Features_sixteensix=
Feature_Extruction([Path,'8_VolumeUp4.wav']);
 Template_MFCC_Features_sixteenseven=
Feature_Extruction([Path,'8_VolumeDown1.wav']);
 Template_MFCC_Features_sixteeneight=
Feature_Extruction([Path,'8_VolumeDown2.wav']);
 Template_MFCC_Features_sixteennine=
Feature_Extruction([Path,'8_VolumeDown3.wav']);
 Template_MFCC_Features_seventeenone=
Feature_Extruction([Path,'8_VolumeDown4.wav']);
 Template_MFCC_Features_seventeentwo=
Feature_Extruction([Path,'8_ChangeChannel1.wav']);
 Template_MFCC_Features_seventeenthree=
Feature_Extruction([Path,'8_ChangeChannel2.wav']);
 Template_MFCC_Features_seventeenfour=
Feature_Extruction([Path,'8_ChangeChannel3.wav']);
 Template_MFCC_Features_seventeenfive=
Feature_Extruction([Path,'8_ChangeChannel4.wav']);
 Template_MFCC_Features_seventeensix=
Feature_Extruction([Path,'9_SwitchOn1.wav']);
 Template_MFCC_Features_seventeenseven=
Feature_Extruction([Path,'9_SwitchOn2.wav']);
 Template_MFCC_Features_seventeeneight=
Feature_Extruction([Path,'9_SwitchOn3.wav']);
 Template_MFCC_Features_seventeennine=
Feature_Extruction([Path,'9_SwitchOn4.wav']);
 Template_MFCC_Features_eightteenone=
Feature_Extruction([Path,'9_SwitchOff1.wav']);
 Template_MFCC_Features_eightteentwo=
Feature_Extruction([Path,'9_SwitchOff2.wav']);
 Template_MFCC_Features_eightteenthree=
Feature_Extruction([Path,'9_SwitchOff3.wav']);
 Template_MFCC_Features_eightteenfour=
Feature_Extruction([Path,'9_SwitchOff4.wav']);
```

```
 Template_MFCC_Features_eightteenfive=
Feature_Extruction([Path,'9_VolumeUp1.wav']);
 Template_MFCC_Features_eightteensix=
Feature_Extruction([Path,'9_VolumeUp2.wav']);
 Template_MFCC_Features_eightteenseven=
Feature_Extruction([Path,'9_VolumeUp3.wav']);
 Template_MFCC_Features_eightteeneight=
Feature_Extruction([Path,'9_VolumeUp4.wav']);
 Template_MFCC_Features_eightteennine=
Feature_Extruction([Path,'9_VolumeDown1.wav']);
 Template_MFCC_Features_nineteenone=
Feature_Extruction([Path,'9_VolumeDown2.wav']);
 Template_MFCC_Features_nineteentwo=
Feature_Extruction([Path,'9_VolumeDown3.wav']);
 Template_MFCC_Features_nineteenthree=
Feature_Extruction([Path,'9_VolumeDown4.wav']);
 Template_MFCC_Features_nineteenfour=
Feature_Extruction([Path,'9_ChangeChannel1.wav']);
 Template_MFCC_Features_nineteenfive=
Feature_Extruction([Path,'9_ChangeChannel2.wav']);
 Template_MFCC_Features_nineteensix=
Feature_Extruction([Path,'9_ChangeChannel3.wav']);
 Template_MFCC_Features_nineteenseven=
Feature_Extruction([Path,'9_ChangeChannel4.wav']);
 Template_MFCC_Features_nineteeneight=
Feature_Extruction([Path,'10_SwitchOn1.wav']);
 Template_MFCC_Features_nineteennine=
Feature_Extruction([Path,'10_SwitchOn2.wav']);
 Template_MFCC_Features_twentyone=
Feature_Extruction([Path,'10_SwitchOn3.wav']);
 Template_MFCC_Features_twentytwo=
Feature_Extruction([Path,'10_SwitchOn4.wav']);
 Template_MFCC_Features_twentythree=
Feature_Extruction([Path,'10_SwitchOff1.wav']);
 Template_MFCC_Features_twentyfour=
Feature_Extruction([Path,'10_SwitchOff2.wav']);
 Template_MFCC_Features_twentyfive=
Feature_Extruction([Path,'10_SwitchOff3.wav']);
 Template_MFCC_Features_twentysix=
Feature_Extruction([Path,'10_SwitchOff4.wav']);
 Template_MFCC_Features_twentyseven=
Feature_Extruction([Path,'10_VolumeUp1.wav']);
 Template_MFCC_Features_twentyeight=
Feature_Extruction([Path,'10_VolumeUp2.wav']);
 Template_MFCC_Features_twentynine=
Feature_Extruction([Path,'10_VolumeUp3.wav']);
 Template_MFCC_Features_thirtyone=
Feature_Extruction([Path,'10_VolumeUp4.wav']);
```

```
 Template_MFCC_Features_thirtytwo=
Feature_Extruction([Path,'10_VolumeDown1.wav']);
 Template_MFCC_Features_thirtythree=
Feature_Extruction([Path,'10_VolumeDown2.wav']);
 Template_MFCC_Features_thirtyfour=
Feature_Extruction([Path,'10_VolumeDown3.wav']);
 Template_MFCC_Features_thirtyfive=
Feature_Extruction([Path,'10_VolumeDown4.wav']);
 Template_MFCC_Features_thirtysix=
Feature_Extruction([Path,'10_ChangeChannel1.wav']);
 Template_MFCC_Features_thirtyseven=
Feature_Extruction([Path,'10_ChangeChannel2.wav']);
 Template_MFCC_Features_thirtyeight=
Feature_Extruction([Path,'10_ChangeChannel3.wav']);
 Template_MFCC_Features_thirtynine=
Feature_Extruction([Path,'10_ChangeChannel4.wav']);
 Template_MFCC_Features_fourtyone=
Feature_Extruction([Path,'11_SwitchOn1.wav']);
 Template_MFCC_Features_fourtytwo=
Feature_Extruction([Path,'11_SwitchOn2.wav']);
 Template_MFCC_Features_fourtythree=
Feature_Extruction([Path,'11_SwitchOn3.wav']);
 Template_MFCC_Features_fourtyfour=
Feature_Extruction([Path,'11_SwitchOn4.wav']);
 Template_MFCC_Features_fourtyfive=
Feature_Extruction([Path,'11_SwitchOff1.wav']);
 Template_MFCC_Features_fourtysix=
Feature_Extruction([Path,'11_SwitchOff2.wav']);
 Template_MFCC_Features_fourtyseven=
Feature_Extruction([Path,'11_SwitchOff3.wav']);
 Template_MFCC_Features_fourtyeight=
Feature_Extruction([Path,'11_SwitchOff4.wav']);
 Template_MFCC_Features_fourtynine=
Feature_Extruction([Path,'11_VolumeUp1.wav']);
 Template_MFCC_Features_fifthtyone=
Feature_Extruction([Path,'11_VolumeUp2.wav']);
 Template_MFCC_Features_fifthtytwo=
Feature_Extruction([Path,'11_VolumeUp3.wav']);
 Template_MFCC_Features_fifthtythree=
Feature_Extruction([Path,'11_VolumeUp4.wav']);
 Template_MFCC_Features_fifthtyfour=
Feature_Extruction([Path,'11_VolumeDown1.wav']);
 Template_MFCC_Features_fifthtyfive=
Feature_Extruction([Path,'11_VolumeDown2.wav']);
 Template_MFCC_Features_fifthtysix=
Feature_Extruction([Path,'11_VolumeDown3.wav']);
 Template_MFCC_Features_fifthtyseven=
Feature_Extruction([Path,'11_VolumeDown4.wav']);
```

```
 Template_MFCC_Features_fifthtyeight=
Feature_Extruction([Path,'11_ChangeChannel1.wav']);
 Template_MFCC_Features_fifthtynine=
Feature_Extruction([Path,'11_ChangeChannel2.wav']);
 Template_MFCC_Features_sixtyone=
Feature_Extruction([Path,'11_ChangeChannel3.wav']);
 Template_MFCC_Features_sixtytwo=
Feature_Extruction([Path,'11_ChangeChannel4.wav']);
 %save Templates.mat
 save Templates_data.mat
 clear path

function [Temp_F,Temp_N]=SelectNextTemplate(No)
   % Select the next template and return it's name (Temp_N) and feature vectors
(Temp_F)

   load Templates_data.mat;
   %load Templates.mat;

   switch(No)
      case {1}
         Temp_F=Template_MFCC_Features_zero;
         Temp_N='1_SwitchOn1';
      case {2}
         Temp_F=Template_MFCC_Features_one;
         Temp_N='1_SwitchOn2';
      case {3}
         Temp_F=Template_MFCC_Features_two;
         Temp_N='1_SwitchOn3';
      case {4}
         Temp_F=Template_MFCC_Features_three;
         Temp_N='1_SwitchOn4';
      case {5}
         Temp_F=Template_MFCC_Features_four;
         Temp_N='1_SwitchOff1';
      case {6}
         Temp_F=Template_MFCC_Features_five;
         Temp_N='1_SwitchOff2';
      case {7}
         Temp_F=Template_MFCC_Features_six;
         Temp_N='1_SwitchOff3';
      case {8}
         Temp_F=Template_MFCC_Features_seven;
         Temp_N='1_SwitchOff4';
      case {9}
         Temp_F=Template_MFCC_Features_eight;
         Temp_N='1_VolumeUp1';
      case {10}
        Temp_F=Template_MFCC_Features_nine;
```

```
    Temp_N='1_VolumeUp2';
 case {11}
     Temp_F=Template_MFCC_Features_ten;
     Temp_N='1_VolumeUp3';
 case {12}
     Temp_F=Template_MFCC_Features_oneone;
     Temp_N='1_VolumeUp4';
 case {13}
     Temp_F=Template_MFCC_Features_onetwo;
     Temp_N='1_VolumeDown1';
 case {14}
     Temp_F=Template_MFCC_Features_onethree;
     Temp_N='1_VolumeDown2';
 case {15}
     Temp_F=Template_MFCC_Features_onefour;
     Temp_N='1_VolumeDown3';
  case {16}
     Temp_F=Template_MFCC_Features_onefive;
     Temp_N='1_VolumeDown4';
  case {17}
     Temp_F=Template_MFCC_Features_onesix;
     Temp_N='1_ChangeChannel1';
  case {18}
     Temp_F=Template_MFCC_Features_oneseven;
     Temp_N='1_ChangeChannel2';
  case {19}
     Temp_F=Template_MFCC_Features_oneeight;
     Temp_N='1_ChangeChanneln3';
  case {20}
     Temp_F=Template_MFCC_Features_onenine;
     Temp_N='1_ChangeChannel4';
case {21}
    Temp_F=Template_MFCC_Features_twoone;
    Temp_N='2_SwitchOn1';
 case {22}
    Temp_F=Template_MFCC_Features_twotwo;
    Temp_N='2_SwitchOn2';
 case {23}
    Temp_F=Template_MFCC_Features_twothree;
    Temp_N='2_SwitchOn3';
 case {24}
    Temp_F=Template_MFCC_Features_twofour;
    Temp_N='2_SwitchOn4';
 case {25}
    Temp_F=Template_MFCC_Features_twofive;
    Temp_N='2_SwitchOff1';
 case {26}
    Temp_F=Template_MFCC_Features_twosix;
    Temp_N='2_SwitchOff2';
```

```
case {27}
   Temp_F=Template_MFCC_Features_twoseven;
   Temp_N='2_SwitchOff3';
case {28}
   Temp_F=Template_MFCC_Features_twoeight;
   Temp_N='2_SwitchOff4';
case {29}
   Temp_F=Template_MFCC_Features_twonine;
   Temp_N='2_VolumeUp1';
case {30}
  Temp_F=Template_MFCC_Features_threeone;
   Temp_N='2_VolumeUp2';
case {31}
    Temp_F=Template_MFCC_Features_threetwo;
    Temp_N='2_VolumeUp3';
case {32}
    Temp_F=Template_MFCC_Features_threethree;
    Temp_N='2_VolumeUp4';
case {33}
    Temp_F=Template_MFCC_Features_threefour;
    Temp_N='2_VolumeDown1';
case {34}
    Temp_F=Template_MFCC_Features_threefive;
    Temp_N='2_VolumeDown2';
case {35}
    Temp_F=Template_MFCC_Features_threesix;
    Temp_N='2_VolumeDown3';
 case {36}
    Temp_F=Template_MFCC_Features_threeseven;
    Temp_N='2_VolumeDown4';
 case {37}
    Temp_F=Template_MFCC_Features_threeeight;
    Temp_N='2_ChangeChannel1';
 case {38}
    Temp_F=Template_MFCC_Features_threenine;
    Temp_N='2_ChangeChannel2';
 case {39}
    Temp_F=Template_MFCC_Features_fourone;
    Temp_N='2_ChangeChannel3';
 case {40}
    Temp_F=Template_MFCC_Features_fourtwo;
    Temp_N='2_ChangeChannel4';
 case {41}
   Temp_F=Template_MFCC_Features_fourthree;
   Temp_N='3_SwitchOn1';
case {42}
   Temp_F=Template_MFCC_Features_fourfour;
   Temp_N='3_SwitchOn2';
case {43}
```

```
    Temp_F=Template_MFCC_Features_fourfive;
    Temp_N='3_SwitchOn3';
case {44}
    Temp_F=Template_MFCC_Features_foursix;
    Temp_N='3_SwitchOn4';
case {45}
    Temp_F=Template_MFCC_Features_fourseven;
    Temp_N='3_SwitchOff1';
case {46}
    Temp_F=Template_MFCC_Features_foureight;
    Temp_N='3_SwitchOff2';
case {47}
    Temp_F=Template_MFCC_Features_fournine;
    Temp_N='3_SwitchOff3';
case {48}
    Temp_F=Template_MFCC_Features_fiveone;
    Temp_N='3_SwitchOff4';
case {49}
    Temp_F=Template_MFCC_Features_fivetwo;
    Temp_N='3_VolumeUp1';
case {50}
   Temp_F=Template_MFCC_Features_fivethree;
    Temp_N='3_VolumeUp2';
case {51}
     Temp_F=Template_MFCC_Features_fivefour;
     Temp_N='3_VolumeUp3';
case {52}
     Temp_F=Template_MFCC_Features_fivefive;
     Temp_N='3_VolumeUp4';
case {53}
     Temp_F=Template_MFCC_Features_fivesix;
     Temp_N='3_VolumeDown1';
case {54}
     Temp_F=Template_MFCC_Features_fiveseven;
     Temp_N='3_VolumeDown2';
case {55}
     Temp_F=Template_MFCC_Features_fiveeight;
     Temp_N='3_VolumeDown3';
 case {56}
     Temp_F=Template_MFCC_Features_fivenine;
     Temp_N='3_VolumeDown4';
 case {57}
     Temp_F = Template_MFCC_Features_sixone;
     Temp_N='3_ChangeChannel1';
 case {58}
     Temp_F=Template_MFCC_Features_sixtwo;
     Temp_N='3_ChangeChannel2';
 case {59}
     Temp_F=Template_MFCC_Features_sixthree;
```

```
    Temp_N='3_ChangeChannel3';
 case {60}
    Temp_F=Template_MFCC_Features_sixfour;
    Temp_N='3_ChangeChannel4';
case {61}
   Temp_F=Template_MFCC_Features_sixfive;
   Temp_N='4_SwitchOn1';
case {62}
   Temp_F=Template_MFCC_Features_sixsix;
   Temp_N='4_SwitchOn2';
case {63}
   Temp_F=Template_MFCC_Features_sixseven;
   Temp_N='4_SwitchOn3';
case {64}
   Temp_F=Template_MFCC_Features_sixeight;
   Temp_N='4_SwitchOn4';
case {65}
   Temp_F=Template_MFCC_Features_sixnine;
   Temp_N='4_SwitchOff1';
case {66}
   Temp_F=Template_MFCC_Features_sevenone;
   Temp_N='4_SwitchOff2';
case {67}
   Temp_F=Template_MFCC_Features_seventwo;
   Temp_N='4_SwitchOff3';
case {68}
   Temp_F=Template_MFCC_Features_seventhree;
   Temp_N='4_SwitchOff4';
case {69}
   Temp_F=Template_MFCC_Features_sevenfour;
   Temp_N='4_VolumeUp1';
case {70}
  Temp_F=Template_MFCC_Features_sevenfive;
   Temp_N='4_VolumeUp2';
case {71}
    Temp_F=Template_MFCC_Features_sevensix;
    Temp_N='4_VolumeUp3';
case {72}
    Temp_F=Template_MFCC_Features_sevenseven;
    Temp_N='4_VolumeUp4';
case {73}
    Temp_F=Template_MFCC_Features_seveneight;
    Temp_N='4_VolumeDown1';
case {74}
    Temp_F=Template_MFCC_Features_sevennine;
    Temp_N='4_VolumeDown2';
case {75}
    Temp_F=Template_MFCC_Features_eightone;
    Temp_N='4_VolumeDown3';
```

```
 case {76}
    Temp_F=Template_MFCC_Features_eighttwo;
    Temp_N='4_VolumeDown4';
 case {77}
    Temp_F = Template_MFCC_Features_eightthree;
    Temp_N='4_ChangeChannel1';
 case {78}
    Temp_F=Template_MFCC_Features_eightfour;
    Temp_N='4_ChangeChannel2';
 case {79}
    Temp_F=Template_MFCC_Features_eightfive;
    Temp_N='4_ChangeChannel3';
 case {80}
    Temp_F=Template_MFCC_Features_eightsix;
    Temp_N='4_ChangeChannel4';

case {81}
   Temp_F=Template_MFCC_Features_eightseven;
   Temp_N='5_SwitchOn1';
case {82}
   Temp_F=Template_MFCC_Features_eighteight;
   Temp_N='5_SwitchOn2';
case {83}
   Temp_F=Template_MFCC_Features_eightnine;
   Temp_N='5_SwitchOn3';
case {84}
   Temp_F=Template_MFCC_Features_nineone;
   Temp_N='5_SwitchOn4';
case {85}
   Temp_F=Template_MFCC_Features_ninetwo;
   Temp_N='5_SwitchOff1';
case {86}
   Temp_F=Template_MFCC_Features_ninethree;
   Temp_N='5_SwitchOff2';
case {87}
   Temp_F=Template_MFCC_Features_ninefour;
   Temp_N='5_SwitchOff3';
case {88}
   Temp_F=Template_MFCC_Features_ninefive;
   Temp_N='5_SwitchOff4';
case {89}
   Temp_F=Template_MFCC_Features_ninesix;
   Temp_N='5_VolumeUp1';
case {90}
  Temp_F=Template_MFCC_Features_nineseven;
   Temp_N='5_VolumeUpa2';
case {91}
    Temp_F=Template_MFCC_Features_nineeight;
    Temp_N='5_VolumeUp3';
```

```
case {92}
    Temp_F=Template_MFCC_Features_ninenine;
    Temp_N='5_VolumeUp4';
case {93}
    Temp_F=Template_MFCC_Features_tenone;
    Temp_N='5_VolumeDown1';
case {94}
    Temp_F=Template_MFCC_Features_tentwo;
    Temp_N='5_VolumeDown2';
case {95}
    Temp_F=Template_MFCC_Features_tenthree;
    Temp_N='5_VolumeDown3';
 case {96}
    Temp_F=Template_MFCC_Features_tenfour;
    Temp_N='5_VolumeDown4';
 case {97}
    Temp_F = Template_MFCC_Features_tenfive;
    Temp_N='5_ChangeChannel1';
 case {98}
    Temp_F=Template_MFCC_Features_tensix;
    Temp_N='5_ChangeChannel2';
 case {99}
    Temp_F=Template_MFCC_Features_tenseven;
    Temp_N='5_ChangeChannel3';
 case {100}
    Temp_F=Template_MFCC_Features_teneight;
    Temp_N='5_ChangeChannel4';
case {101}
   Temp_F=Template_MFCC_Features_tennine;
   Temp_N='6_SwitchOn1';
case {102}
   Temp_F=Template_MFCC_Features_elevenone;
   Temp_N='6_SwitchOn2';
case {103}
   Temp_F=Template_MFCC_Features_eleventwo;
   Temp_N='6_SwitchOn3';
case {104}
   Temp_F=Template_MFCC_Features_eleventhree;
   Temp_N='6_SwitchOn4';
case {105}
   Temp_F=Template_MFCC_Features_elevenfour;
   Temp_N='6_SwitchOff1';
case {106}
   Temp_F=Template_MFCC_Features_elevenfive;
   Temp_N='6_SwitchOff2';
case {107}
   Temp_F=Template_MFCC_Features_elevensix;
   Temp_N='6_SwitchOff3';
case {108}
```

```
    Temp_F=Template_MFCC_Features_elevenseven;
    Temp_N='6_SwitchOff4';
case {109}
    Temp_F=Template_MFCC_Features_eleveneight;
    Temp_N='6_VolumeUp1';
case {110}
   Temp_F=Template_MFCC_Features_elevennine;
    Temp_N='6_VolumeUp2';
case {111}
     Temp_F=Template_MFCC_Features_twelveone;
     Temp_N='6_VolumeUp3';
case {112}
     Temp_F=Template_MFCC_Features_twelvetwo;
     Temp_N='6_VolumeUp4';
case {113}
     Temp_F=Template_MFCC_Features_twelvethree;
     Temp_N='6_VolumeDown1';
case {114}
     Temp_F=Template_MFCC_Features_twelvefour;
     Temp_N='6_VolumeDown2';
case {115}
     Temp_F=Template_MFCC_Features_twelvefive;
     Temp_N='6_VolumeDown3';
 case {116}
     Temp_F=Template_MFCC_Features_twelvesix;
     Temp_N='6_VolumeDown4';
 case {117}
     Temp_F = Template_MFCC_Features_twelveseven;
     Temp_N='6_ChangeChannel1';
 case {118}
     Temp_F=Template_MFCC_Features_twelveeight;
     Temp_N='6_ChangeChannel2';
 case {119}
     Temp_F=Template_MFCC_Features_twelvenine;
     Temp_N='6_ChangeChannel3';
 case {120}
     Temp_F=Template_MFCC_Features_thirteenone;
     Temp_N='6_ChangeChannel4';
case {121}
    Temp_F=Template_MFCC_Features_thirteentwo;
    Temp_N='7_SwitchOn1';
case {122}
    Temp_F=Template_MFCC_Features_thirteenthree;
    Temp_N='7_SwitchOn2';
case {123}
    Temp_F=Template_MFCC_Features_thirteenfour;
    Temp_N='7_SwitchOn3';
case {124}
    Temp_F=Template_MFCC_Features_thirteenfive;
```

```
    Temp_N='7_SwitchOn4';
case {125}
    Temp_F=Template_MFCC_Features_thirteensix;
    Temp_N='7_SwitchOff1';
case {126}
    Temp_F=Template_MFCC_Features_thirteenseven;
    Temp_N='7_SwitchOff2';
case {127}
    Temp_F=Template_MFCC_Features_thirteeneight;
    Temp_N='7_SwitchOff3';
case {128}
    Temp_F=Template_MFCC_Features_thirteennine;
    Temp_N='7_SwitchOff4';
case {129}
    Temp_F=Template_MFCC_Features_fourteenone;
    Temp_N='7_VolumeUp1';
case {130}
   Temp_F=Template_MFCC_Features_fourteentwo;
    Temp_N='7_VolumeUp2';
case {131}
     Temp_F=Template_MFCC_Features_fourteenthree;
     Temp_N='7_VolumeUp3';
case {132}
     Temp_F=Template_MFCC_Features_fourteenfour;
     Temp_N='7_VolumeUp4';
case {133}
     Temp_F=Template_MFCC_Features_fourteenfive;
     Temp_N='7_VolumeDown1';
case {134}
     Temp_F=Template_MFCC_Features_fourteensix;
     Temp_N='7_VolumeDown2';
case {135}
     Temp_F=Template_MFCC_Features_fourteenseven;
     Temp_N='7_VolumeDown3';
 case {136}
     Temp_F=Template_MFCC_Features_fourteeneight;
     Temp_N='7_VolumeDowna4';
 case {137}
     Temp_F = Template_MFCC_Features_fourteennine;
     Temp_N='7_ChangeChannel1';
 case {138}
     Temp_F=Template_MFCC_Features_fifthteenone;
     Temp_N='7_ChangeChannel2';
 case {139}
     Temp_F=Template_MFCC_Features_fifthteentwo;
     Temp_N='7_ChangeChannel3';
 case {140}
     Temp_F=Template_MFCC_Features_fifthteenthree;
     Temp_N='7_ChangeChannel4';
```

```
case {141}
   Temp_F=Template_MFCC_Features_fifthteenfour;
   Temp_N='8_SwitchOn1';
case {142}
   Temp_F=Template_MFCC_Features_fifthteenfive;
   Temp_N='8_SwitchOn2';
case {143}
   Temp_F=Template_MFCC_Features_fifthteensix;
   Temp_N='8_SwitchOn3';
case {144}
   Temp_F=Template_MFCC_Features_fifthteenseven;
   Temp_N='8_SwitchOn4';
case {145}
   Temp_F=Template_MFCC_Features_fifthteeneight;
   Temp_N='8_SwitchOff';
case {146}
   Temp_F=Template_MFCC_Features_fifthteennine;
   Temp_N='8_SwitchOff2';
case {147}
   Temp_F=Template_MFCC_Features_sixteenone;
   Temp_N='8_SwitchOff3';
case {148}
   Temp_F=Template_MFCC_Features_sixteentwo;
   Temp_N='8_SwitchOff4';
case {149}
   Temp_F=Template_MFCC_Features_sixteenthree;
   Temp_N='8_VolumeUp1';
case {150}
  Temp_F=Template_MFCC_Features_sixteenfour;
   Temp_N='8_VolumeUp2';
case {151}
    Temp_F=Template_MFCC_Features_sixteenfive;
    Temp_N='8_VolumeUp3';
case {152}
    Temp_F=Template_MFCC_Features_sixteensix;
    Temp_N='8_VolumeUp4';
case {153}
    Temp_F=Template_MFCC_Features_sixteenseven;
    Temp_N='8_VolumeDown1';
case {154}
    Temp_F=Template_MFCC_Features_sixteeneight;
    Temp_N='8_VolumeDown2';
case {155}
    Temp_F=Template_MFCC_Features_sixteennine;
    Temp_N='8_VolumeDown3';
 case {156}
    Temp_F=Template_MFCC_Features_seventeenone;
    Temp_N='8_VolumeDown4';
 case {157}
```

```
    Temp_F = Template_MFCC_Features_seventeentwo;
    Temp_N='8_ChangeChannel1';
 case {158}
    Temp_F=Template_MFCC_Features_seventeenthree;
    Temp_N='8_ChangeChannel2';
 case {159}
    Temp_F=Template_MFCC_Features_seventeenfour;
    Temp_N='8_ChangeChannel3';
 case {160}
    Temp_F=Template_MFCC_Features_seventeenfive;
    Temp_N='8_ChangeChannel4';
case {161}
   Temp_F=Template_MFCC_Features_seventeensix;
   Temp_N='9_SwitchOn1';
case {162}
   Temp_F=Template_MFCC_Features_seventeenseven;
   Temp_N='9_SwitchOn2';
case {163}
   Temp_F=Template_MFCC_Features_seventeeneight;
   Temp_N='9_SwitchOn3';
case {164}
   Temp_F=Template_MFCC_Features_seventeennine;
   Temp_N='9_SwitchOn4';
case {165}
   Temp_F=Template_MFCC_Features_eightteenone;
   Temp_N='9_SwitchOff1';
case {166}
   Temp_F=Template_MFCC_Features_eightteentwo;
   Temp_N='9_SwitchOff2';
case {167}
   Temp_F=Template_MFCC_Features_eightteenthree;
   Temp_N='9_SwitchOff3';
case {168}
   Temp_F=Template_MFCC_Features_eightteenfour;
   Temp_N='9_SwitchOff4';
case {169}
   Temp_F=Template_MFCC_Features_eightteenfive;
   Temp_N='9_VolumeUp1';
case {170}
  Temp_F=Template_MFCC_Features_eightteensix;
   Temp_N='9_VolumeUp2';
case {171}
    Temp_F=Template_MFCC_Features_eightteenseven;
    Temp_N='9_VolumeUp3';
case {172}
    Temp_F=Template_MFCC_Features_eightteeneight;
    Temp_N='9_VolumeUp4';
case {173}
    Temp_F=Template_MFCC_Features_eightteennine;
```

```
    Temp_N='9_VolumeDown1';
case {174}
    Temp_F=Template_MFCC_Features_nineteenone;
    Temp_N='9_VolumeDown2';
case {175}
    Temp_F=Template_MFCC_Features_nineteentwo;
    Temp_N='9_VolumeDown3';
 case {176}
    Temp_F=Template_MFCC_Features_nineteenthree;
    Temp_N='9_VolumeDown4';
 case {177}
    Temp_F = Template_MFCC_Features_nineteenfour;
    Temp_N='9_ChangeChannel1';
 case {178}
    Temp_F=Template_MFCC_Features_nineteenfive;
    Temp_N='9_ChangeChannel2';
 case {179}
    Temp_F=Template_MFCC_Features_nineteensix;
    Temp_N='9_ChangeChannel3';
 case {180}
    Temp_F=Template_MFCC_Features_nineteenseven;
    Temp_N='9_ChangeChannel4';
case {181}
   Temp_F=Template_MFCC_Features_nineteeneight;
   Temp_N='10_SwitchOn1';
case {182}
   Temp_F=Template_MFCC_Features_nineteennine;
   Temp_N='10_SwitchOn2';
case {183}
   Temp_F=Template_MFCC_Features_twentyone;
   Temp_N='10_SwitchOn3';
case {184}
   Temp_F=Template_MFCC_Features_twentytwo;
   Temp_N='10_SwitchOn4';
case {185}
   Temp_F=Template_MFCC_Features_twentythree;
   Temp_N='10_SwitchOff1';
case {186}
   Temp_F=Template_MFCC_Features_twentyfour;
   Temp_N='10_SwitchOff2';
case {187}
   Temp_F=Template_MFCC_Features_twentyfive;
   Temp_N='10_SwitchOff3';
case {188}
   Temp_F=Template_MFCC_Features_twentysix;
   Temp_N='10_SwitchOff4';
case {189}
   Temp_F=Template_MFCC_Features_twentyseven;
   Temp_N='10_VolumeUp1';
```

```
case {190}
   Temp_F=Template_MFCC_Features_twentyeight;
    Temp_N='10_VolumeUp2';
case {191}
     Temp_F=Template_MFCC_Features_twentynine;
     Temp_N='10_VolumeUp3';
case {192}
     Temp_F=Template_MFCC_Features_thirtyone;
     Temp_N='10_VolumeUp4';
case {193}
     Temp_F=Template_MFCC_Features_thirtytwo;
     Temp_N='10_VolumeDown1';
case {194}
     Temp_F=Template_MFCC_Features_thirtythree;
     Temp_N='10_VolumeDown2';
case {195}
     Temp_F=Template_MFCC_Features_thirtyfour;
     Temp_N='10_VolumeDown3';
 case {196}
     Temp_F=Template_MFCC_Features_thirtyfive;
     Temp_N='10_VolumeDown4';
 case {197}
     Temp_F = Template_MFCC_Features_thirtysix;
     Temp_N='10_ChangeChannel1';
 case {198}
     Temp_F=Template_MFCC_Features_thirtyseven;
     Temp_N='10_ChangeChannel2';
 case {199}
     Temp_F=Template_MFCC_Features_thirtyeight;
     Temp_N='10_ChangeChannel3';
 case {200}
     Temp_F=Template_MFCC_Features_thirtynine;
     Temp_N='10_ChangeChannel4';
case {201}
    Temp_F=Template_MFCC_Features_fourtyone;
    Temp_N='11_SwitchOn1';
case {202}
    Temp_F=Template_MFCC_Features_fourtytwo;
    Temp_N='11_SwitchOn2';
case {203}
    Temp_F=Template_MFCC_Features_fourtythree;
    Temp_N='11_SwitchOn3';
case {204}
    Temp_F=Template_MFCC_Features_fourtyfour;
    Temp_N='11_SwitchOn4';
case {205}
    Temp_F=Template_MFCC_Features_fourtyfive;
    Temp_N='11_SwitchOff1';
case {206}
```

```
        Temp_F=Template_MFCC_Features_fourtysix;
        Temp_N='11_SwitchOff2';
    case {207}
        Temp_F=Template_MFCC_Features_fourtyseven;
        Temp_N='11_SwitchOff3';
    case {208}
        Temp_F=Template_MFCC_Features_fourtyeight;
        Temp_N='11_SwitchOff4';
    case {209}
        Temp_F=Template_MFCC_Features_fourtynine;
        Temp_N='11_VolumeUp1';
    case {210}
       Temp_F=Template_MFCC_Features_fifthtyone;
        Temp_N='11_VolumeUp2';
    case {211}
         Temp_F=Template_MFCC_Features_fifthtytwo;
         Temp_N='11_VolumeUp3';
    case {212}
         Temp_F=Template_MFCC_Features_fifthtythree;
         Temp_N='11_VolumeUp4';
    case {213}
         Temp_F=Template_MFCC_Features_fifthtyfour;
         Temp_N='11_VolumeDown1';
    case {214}
         Temp_F=Template_MFCC_Features_fifthtyfive;
         Temp_N='11_VolumeDown2';
    case {215}
         Temp_F=Template_MFCC_Features_fifthtysix;
         Temp_N='11_VolumeDown3';
     case {216}
         Temp_F=Template_MFCC_Features_fifthtyseven;
         Temp_N='11_VolumeDown4';
     case {217}
         Temp_F = Template_MFCC_Features_fifthtyeight;
         Temp_N='11_ChangeChannel1';
     case {218}
         Temp_F=Template_MFCC_Features_fifthtynine;
         Temp_N='11_ChangeChannel2';
     case {219}
         Temp_F=Template_MFCC_Features_sixtyone;
         Temp_N='11_ChangeChannel3';
     case {220}
         Temp_F=Template_MFCC_Features_sixtytwo;
         Temp_N='11_ChangeChannel4';

        otherwise
        error;
end
```

2) Malay words

```
% Dynamic Time Warping (DTW)
 % Extructing features of Templates and save them.

 clear all;
 close all;
% clf;
 clc;
 %Path='D:\MYProject\Voice_DTW-Versi-Linda\Data\';
 Path='Data\';
Template_List=['1_Buka','1_Tutup','1_KuatkanSuara','1_PerlahankanSuara','1_Tukar
Siaran','2_Buka','2_Tutup','2_KuatkanSuara','2_PerlahankanSuara','2_TukarSiaran','3_
Buka','3_Tutup','3_KuatkanSuara','3_PerlahankanSuara','3_TukarSiaran''4_Buka','4_T
utup','4_KuatkanSuara','4_PerlahankanSuara','4_TukarSiaran',
'5_Buka','5_Tutup','5_KuatkanSuara','5_PerlahankanSuara','5_TukarSiaran','6_Buka','
6_Tutup','6_KuatkanSuara','6_PerlahankanSuara','6_TukarSiaran','7_Buka','7_Tutup','
7_KuatkanSuara','7_PerlahankanSuara','7_TukarSiaran','8_Buka','8_Tutup','8_Kuatka
nSuara','8_PerlahankanSuara','8_TukarSiaran','9_Buka','9_Tutup','9_KuatkanSuara','9
_PerlahankanSuara','9_TukarSiaran','10_Buka','10_Tutup','10_KuatkanSuara','10_Perl
ahankanSuara','10_TukarSiaran','11_Buka','11_Tutup','11_KuatkanSuara','11_Perlaha
nkanSuara','11_TukarSiaran','12_Buka','12_Tutup','12_KuatkanSuara','12_Perlahanka
nSuara','12_TukarSiaran'];

 Template_MFCC_Features_zero=
CMS_Normalization(Feature_Extruction([Path,'1_Buka.wav']));
 Template_MFCC_Features_one=
CMS_Normalization(Feature_Extruction([Path,'1_Tutup.wav']));
 Template_MFCC_Features_two=
CMS_Normalization(Feature_Extruction([Path,'1_KuatkanSuara.wav']));
 Template_MFCC_Features_three=
CMS_Normalization(Feature_Extruction([Path,'1_PerlahankanSuara.wav']));
 Template_MFCC_Features_four=
CMS_Normalization(Feature_Extruction([Path,'1_TukarSiaran.wav']));
 Template_MFCC_Features_five=
CMS_Normalization(Feature_Extruction([Path,'2_Buka.wav']));
 Template_MFCC_Features_six=
CMS_Normalization(Feature_Extruction([Path,'2_Tutup.wav']));
 Template_MFCC_Features_seven=
CMS_Normalization(Feature_Extruction([Path,'2_KuatkanSuara.wav']));
 Template_MFCC_Features_eight=
CMS_Normalization(Feature_Extruction([Path,'2_PerlahankanSuara.wav']));
 Template_MFCC_Features_nine=
CMS_Normalization(Feature_Extruction([Path,'2_TukarSiaran.wav']));
 Template_MFCC_Features_ten=
CMS_Normalization(Feature_Extruction([Path,'3_Buka.wav']));
 Template_MFCC_Features_oneone=
CMS_Normalization(Feature_Extruction([Path,'3_Tutup.wav']));
```

```
 Template_MFCC_Features_onetwo=
CMS_Normalization(Feature_Extruction([Path,'3_KuatkanSuara.wav']));
 Template_MFCC_Features_onethree=
CMS_Normalization(Feature_Extruction([Path,'3_PerlahankanSuara.wav']));
 Template_MFCC_Features_onefour=
CMS_Normalization(Feature_Extruction([Path,'3_TukarSiaran.wav']));
 Template_MFCC_Features_onefive=
CMS_Normalization(Feature_Extruction([Path,'4_Buka.wav']));
 Template_MFCC_Features_onesix=
CMS_Normalization(Feature_Extruction([Path,'4_Tutup.wav']));
 Template_MFCC_Features_oneseven=
CMS_Normalization(Feature_Extruction([Path,'4_KuatkanSuara.wav']));
 Template_MFCC_Features_oneeight=
CMS_Normalization(Feature_Extruction([Path,'4_PerlahankanSuara.wav']));
 Template_MFCC_Features_onenine=
CMS_Normalization(Feature_Extruction([Path,'4_TukarSiaran.wav']));
 Template_MFCC_Features_twoone=
CMS_Normalization(Feature_Extruction([Path,'5_Buka.wav']));
 Template_MFCC_Features_twotwo=
CMS_Normalization(Feature_Extruction([Path,'5_Tutup.wav']));
 Template_MFCC_Features_twothree=
CMS_Normalization(Feature_Extruction([Path,'5_KuatkanSuara.wav']));
 Template_MFCC_Features_twofour=
CMS_Normalization(Feature_Extruction([Path,'5_PerlahankanSuara.wav']));
 Template_MFCC_Features_twofive=
CMS_Normalization(Feature_Extruction([Path,'5_TukarSiaran.wav']));
 Template_MFCC_Features_twosix=
CMS_Normalization(Feature_Extruction([Path,'6_Buka.wav']));
 Template_MFCC_Features_twoseven=
CMS_Normalization(Feature_Extruction([Path,'6_Tutup.wav']));
 Template_MFCC_Features_twoeight=
CMS_Normalization(Feature_Extruction([Path,'6_KuatkanSuara.wav']));
 Template_MFCC_Features_twonine=
CMS_Normalization(Feature_Extruction([Path,'6_PerlahankanSuara.wav']));
 Template_MFCC_Features_threeone=
CMS_Normalization(Feature_Extruction([Path,'6_TukarSiaran.wav']));
 Template_MFCC_Features_threetwo=
CMS_Normalization(Feature_Extruction([Path,'7_Buka.wav']));
 Template_MFCC_Features_threethree=
CMS_Normalization(Feature_Extruction([Path,'7_Tutup.wav']));
 Template_MFCC_Features_threefour=
CMS_Normalization(Feature_Extruction([Path,'7_KuatkanSuara.wav']));
 Template_MFCC_Features_threefive=
CMS_Normalization(Feature_Extruction([Path,'7_PerlahankanSuara.wav']));
 Template_MFCC_Features_threesix=
CMS_Normalization(Feature_Extruction([Path,'7_TukarSiaran.wav']));
 Template_MFCC_Features_threeseven=
CMS_Normalization(Feature_Extruction([Path,'8_Buka.wav']));
```

```
 Template_MFCC_Features_threeeight=
CMS_Normalization(Feature_Extruction([Path,'8_Tutup.wav']));
 Template_MFCC_Features_threenine=
CMS_Normalization(Feature_Extruction([Path,'8_KuatkanSuara.wav']));
 Template_MFCC_Features_fourone=
CMS_Normalization(Feature_Extruction([Path,'8_PerlahankanSuara.wav']));
 Template_MFCC_Features_fourtwo=
CMS_Normalization(Feature_Extruction([Path,'8_TukarSiaran.wav']));
 Template_MFCC_Features_fourthree=
CMS_Normalization(Feature_Extruction([Path,'9_Buka.wav']));
 Template_MFCC_Features_fourfour=
CMS_Normalization(Feature_Extruction([Path,'9_Tutup.wav']));
 Template_MFCC_Features_fourfive=
CMS_Normalization(Feature_Extruction([Path,'9_KuatkanSuara.wav']));
 Template_MFCC_Features_foursix=
CMS_Normalization(Feature_Extruction([Path,'9_PerlahankanSuara.wav']));
 Template_MFCC_Features_fourseven=
CMS_Normalization(Feature_Extruction([Path,'9_TukarSiaran.wav']));
 Template_MFCC_Features_foureight=
CMS_Normalization(Feature_Extruction([Path,'10_Buka.wav']));
 Template_MFCC_Features_fournine=
CMS_Normalization(Feature_Extruction([Path,'10_Tutup.wav']));
 Template_MFCC_Features_fiveone=
CMS_Normalization(Feature_Extruction([Path,'10_KuatkanSuara.wav']));
 Template_MFCC_Features_fivetwo=
CMS_Normalization(Feature_Extruction([Path,'10_PerlahankanSuara.wav']));
 Template_MFCC_Features_fivethree=
CMS_Normalization(Feature_Extruction([Path,'10_TukarSiaran.wav']));
 Template_MFCC_Features_fivefour=
CMS_Normalization(Feature_Extruction([Path,'11_Buka.wav']));
 Template_MFCC_Features_fivefive=
CMS_Normalization(Feature_Extruction([Path,'11_Tutup.wav']));
 Template_MFCC_Features_fivesix=
CMS_Normalization(Feature_Extruction([Path,'11_KuatkanSuara.wav']));
 Template_MFCC_Features_fiveseven=
CMS_Normalization(Feature_Extruction([Path,'11_PerlahankanSuara.wav']));
 Template_MFCC_Features_fiveeight=
CMS_Normalization(Feature_Extruction([Path,'11_TukarSiaran.wav']));
 Template_MFCC_Features_fivenine=
CMS_Normalization(Feature_Extruction([Path,'12_Buka.wav']));
 Template_MFCC_Features_sixone=
CMS_Normalization(Feature_Extruction([Path,'12_Tutup.wav']));
 Template_MFCC_Features_sixtwo=
CMS_Normalization(Feature_Extruction([Path,'12_KuatkanSuara.wav']));
 Template_MFCC_Features_sixthree=
CMS_Normalization(Feature_Extruction([Path,'12_PerlahankanSuara.wav']));
 Template_MFCC_Features_sixfour=
CMS_Normalization(Feature_Extruction([Path,'12_TukarSiaran.wav']));
```

```
%save Templates.mat
save Templates_data.mat

clear path

function [Temp_F,Temp_N]=SelectNextTemplate(No)
% Select the next template and return it's name (Temp_N) and feature vectors
(Temp_F)

   load Templates_data1.mat;
   %load Templates.mat;

   switch(No)
      case {1}
         Temp_F=Template_MFCC_Features_zero;
         Temp_N='1_Buka';
      case {2}
         Temp_F=Template_MFCC_Features_one;
         Temp_N='1_Tutup';
      case {3}
         Temp_F=Template_MFCC_Features_two;
         Temp_N='1_KuatkanSuara';
      case {4}
         Temp_F=Template_MFCC_Features_three;
         Temp_N='1_PerlahankanSuara';
      case {5}
         Temp_F=Template_MFCC_Features_four;
         Temp_N='1_TukarSiaran';
      case {6}
         Temp_F=Template_MFCC_Features_five;
         Temp_N='2_Buka';
      case {7}
         Temp_F=Template_MFCC_Features_six;
         Temp_N='2_Tutup';
      case {8}
         Temp_F=Template_MFCC_Features_seven;
         Temp_N='2_KuatkanSuara';
      case {9}
         Temp_F=Template_MFCC_Features_eight;
         Temp_N='2_PerlahankanSuara';
      case {10}
        Temp_F=Template_MFCC_Features_nine;
         Temp_N='2_TukarSiaran';
      case {11}
          Temp_F=Template_MFCC_Features_ten;
          Temp_N='3_Buka';
      case {12}
          Temp_F=Template_MFCC_Features_oneone;
          Temp_N='3_Tutup';
```

```
case {13}
    Temp_F=Template_MFCC_Features_onetwo;
    Temp_N='3_KuatkanSuara';
case {14}
    Temp_F=Template_MFCC_Features_onethree;
    Temp_N='3_PerlahankanSuara';
case {15}
    Temp_F=Template_MFCC_Features_onefour;
    Temp_N='3_TukarSiaran';
 case {16}
    Temp_F=Template_MFCC_Features_onefive;
    Temp_N='4_Buka';
 case {17}
    Temp_F=Template_MFCC_Features_onesix;
    Temp_N='4_Tutup';
 case {18}
    Temp_F=Template_MFCC_Features_oneseven;
    Temp_N='4_KuatkanSuara';
 case {19}
    Temp_F=Template_MFCC_Features_oneeight;
    Temp_N='4_PerlahankanSuara';
 case {20}
    Temp_F=Template_MFCC_Features_onenine;
    Temp_N='4_TukarSiaran';
case {21}
    Temp_F=Template_MFCC_Features_twoone;
    Temp_N='5_Buka';
case {22}
    Temp_F=Template_MFCC_Features_twotwo;
   Temp_N='5_Tutup';
 case {23}
   Temp_F=Template_MFCC_Features_twothree;
    Temp_N='5_KuatkanSuara';
 case {24}
    Temp_F=Template_MFCC_Features_twofour;
    Temp_N='5_PerlahankanSuara';
 case {25}
    Temp_F=Template_MFCC_Features_twofive;
   Temp_N='5_TukarSiaran';
 case {26}
    Temp_F=Template_MFCC_Features_twosix;
    Temp_N='6_Buka';
case {27}
    Temp_F=Template_MFCC_Features_twoseven;
   Temp_N='6_Tutup';
 case {28}
   Temp_F=Template_MFCC_Features_twoeight;
    Temp_N='6_KuatkanSuara';
 case {29}
```

```
    Temp_F=Template_MFCC_Features_twonine;
    Temp_N='6_PerlahankanSuara';
 case {30}
    Temp_F=Template_MFCC_Features_threeone;
   Temp_N='6_TukarSiaran';
case {31}
    Temp_F=Template_MFCC_Features_threetwo;
    Temp_N='7_Buka';
case {32}
    Temp_F=Template_MFCC_Features_threethree;
   Temp_N='7_Tutup';
 case {33}
   Temp_F=Template_MFCC_Features_threefour;
    Temp_N='7_KuatkanSuara';
 case {34}
    Temp_F=Template_MFCC_Features_threefive;
    Temp_N='7_PerlahankanSuara';
 case {35}
    Temp_F=Template_MFCC_Features_threesix;
   Temp_N='7_TukarSiaran';
case {36}
    Temp_F=Template_MFCC_Features_threeseven;
    Temp_N='8_Buka';
case {37}
    Temp_F=Template_MFCC_Features_threeeight;
   Temp_N='8_Tutup';
 case {38}
   Temp_F=Template_MFCC_Features_threenine;
    Temp_N='8_KuatkanSuara';
 case {39}
    Temp_F=Template_MFCC_Features_fourone;
    Temp_N='8_PerlahankanSuara';
 case {40}
    Temp_F=Template_MFCC_Features_fourtwo;
   Temp_N='8_TukarSi
case {41}
    Temp_F=Template_MFCC_Features_fourthree;
   Temp_N='9_Buka';
case {42}
    Temp_F=Template_MFCC_Features_fourfour;
   Temp_N='9_Tutup';
case {43}
    Temp_F=Template_MFCC_Features_fourfive;
   Temp_N='9_KuatkanSuara';
case {44}
    Temp_F=Template_MFCC_Features_foursix;
   Temp_N='9_PerlahankanSuara';
case {45}
    Temp_F=Template_MFCC_Features_fourseven;
```

```
        Temp_N='9_TukarSiaran';
    case {46}
        Temp_F=Template_MFCC_Features_foureight;
        Temp_N='10_Buka';
    case {47}
        Temp_F=Template_MFCC_Features_fournine;
        Temp_N='10_Tutup';
    case {48}
        Temp_F=Template_MFCC_Features_fiveone;
        Temp_N='10_KuatkanSuara';
    case {49}
        Temp_F=Template_MFCC_Features_fivetwo;
        Temp_N='10_PerlahankanSuara';
    case {50}
        Temp_F=Template_MFCC_Features_fivethree;
        Temp_N='10_TukarSiaran';
    case {51}
        Temp_F=Template_MFCC_Features_fivefour;
        Temp_N='11_Buka';
    case {52}
        Temp_F=Template_MFCC_Features_fivefive;
        Temp_N='11_Tutup';
    case {53}
        Temp_F=Template_MFCC_Features_fivesix;
        Temp_N='11_KuatkanSuara';
    case {54}
        Temp_F=Template_MFCC_Features_fiveseven;
        Temp_N='11_PerlahankanSuara';
    case {55}
        Temp_F=Template_MFCC_Features_fiveeight;
        Temp_N='11_TukarSiaran';
    case {56}
        Temp_F=Template_MFCC_Features_fivenine;
        Temp_N='12_Buka';
    case {57}
        Temp_F=Template_MFCC_Features_sixone;
        Temp_N='12_Tutup';
    case {58}
        Temp_F=Template_MFCC_Features_sixtwo;
        Temp_N='12_KuatkanSuara';
    case {59}
        Temp_F=Template_MFCC_Features_sixthree;
        Temp_N='12_PerlahankanSuara';
    case {60}
        Temp_F=Template_MFCC_Features_sixfour;
        Temp_N='12_TukarSiaran';
    otherwise
        error;
end
```