

**THE DESIGN AND DEVELOPMENT OF A SYSTEM FOR
CONTROLLING AUTOMOTIVE FUNCTIONS USING
SPEECH RECOGNITION**

**(REKABENTUK DAN PEMBANGUNAN SEBUAH SISTEM UNTUK
PENGENDALIAN FUNGSI AUTOMOTIF MENGGUNAKAN
PENGECAMAN SUARA)**

ABD MANAN BIN AHMAD

**RESEARCH VOT NO:
74066**

**Jabatan Kejuruteraan Perisian
Fakulti Sains Komputer dan Sistem Maklumat
Universiti Teknologi Malaysia**

2006

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : THE DESIGN AND DEVELOPMENT OF A SYSTEM FOR
CONTROLLING AUTOMOTIVE FUNCTIONS USING
SPEECH RECOGNITION

Saya ABD MANAN BIN AHMAD
(HURUF BESAR)


Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.

4. * Sila tandakan (/)

- | | | |
|-------------------------------------|--------------|---|
| <input type="checkbox"/> | SULIT | (Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972). |
| <input type="checkbox"/> | TERHAD | (Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan). |
| <input type="checkbox"/> | TIDAK TERHAD | |
| <input checked="" type="checkbox"/> | | |

TANDATANGAN KETUA PENYELIDIK


PROF. MADYA ABD MANAN AHMAD
Ketua Projek Vot: 74066
Fakulti Sains Komputer & Sistem Makluma:
Universiti Teknologi Malaysia Skudai.

Nama & Cop Ketua Penyelidik

Tarikh : 1.11.2006

CATATAN : * Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan

ABSTRACT

As car manufacturer installed more electronic control interfaces like Wifi, Bluetooth or infrared systems, the maneuverability and accessibility of the automobile itself is enhanced. We opted for personal digital assistant (PDA) which has the capabilities of a computer yet compact enough to be mounted on car's dashboard. The PDA is equipped with automatic speech recognition (ASR) system, thus providing the mean for activating devices via voice. Other benefits gained from using PDA includes easy upgradeability of the ASR engine and resolving the portability issue as consumers may own more than a single car. Apart from the embedded Artificial Neural Network (ANN) based ASR engine, we developed a prototype engine denoted as Support Vector Machine-Dynamic Shifting Window (SVM-DSW) to accommodate speaker independent mode. SVM-DSW is targeted for applications that demand accuracy and reliability. Accuracy gets top priority in high risk tasks such as driving or piloting, surgical procedures, etc. where the slightest error gives disastrous consequences. Consistency is also of paramount prerequisite because the accuracy has to be reproducible time and time again without failure. SVM-DSW has both of these qualities as well as being low in computational cost (using whole word recognition unit and embedded grammar rule) allows it to be ported into Very Large Scale Integration (VLSI) technology. Voice activated household appliances could also benefit from such integration.

ABSTRAK

Sebagai pegilang sesebuah kereta, ia menginstalasikan lebih kepada antaramuka pengawalan elektronik seperti *Wifi*, *Bluetooth* atau sistem *infrared* dan secara tidak langsung pengendalian kereta tersebut semakin meningkat. Kami memilih *Personal Digital Assistant* (PDA) yang mana ia memiliki kebolehan sesebuah komputer yang cukup kompak untuk diletakkan pada papan pemuka kereta. PDA tersebut dilengkapi dengan pengecaman suara automatik atau *Automatic Speech Recognition* (ASR) sistem. Dengan itu ia menyediakan pengaktifan alat peranti menerusi penggunaan suara. Kelebihan lain yang boleh diperolehi dengan menggunakan PDA ini termasuklah kebolehpayaan menaik-taraf enjin ASR yang mudah dan penyelesaian kepada isu peranti yang mudah-alih kerana berkemungkinan pengguna inginkan pemilikan lebih dari sebuah kereta. Berbeza dengan enjin ASR berasaskan *Artificial Neural Network* (ANN), kami membangunkan enjin prototaip yang dinamakan *Support Vector Machine-Dynamic Shifting Window* (SVM-DSW) untuk menyediakan mod bagi penutur bebas. SVM-DSW mensasarkan aplikasi yang berkehendakan kepada ketepatan dan kebolehpercayaan. Ketepatan diletakkan ke tahap keutamaan yang tertinggi dalam tugas yang berisiko seperti memandu kereta atau mengemudi kapal terbang, prosedur pembedahan dan sebagainya dimana kesilapan kecil akan membawa kepada kecelakaan. Kekonsistenan juga adalah kewajiban yang paling utama kerana ketepatan dihasilkan berulang kali masa demi masa tanpa kegagalan. SVM-DSW mempunyai kedua-dua kualiti ini selain menjimatkan kos pengiraan pengkomputeran (menggunakan seluruh unit pengecaman perkataan dan peraturan tatabahasa) yang membenarkan ia di kelompokkan ke dalam

teknologi *Very Large Scale Integration* (VLSI). Pengaktifan suara bagi kelengkapan rumah juga boleh mendapat faedah daripada perintegrasian tersebut.

CONTENTS

NO	TITLE	PAGE
	ABSTRACT	i
	ABSTRAK	ii
	CONTENTS	iii
	LIST OF FIGURES	vii
	LIST OF TABLES	ix
	LIST OF ABBREVIATIONS	x
	LIST OF APPENDICES	xii
CHAPTER I	PROJECT OVERVIEW	
	1.1 Introduction	1
	1.2 Background	2
	1.3 Problem Statement	2
	1.4 Aim	4
	1.5 Project Objective	4
	1.6 Scope Project	4
	1.7 Thesis Arrangement	6
CHAPTER II	LITERITURE REVIEW	
	2.1 Introduction	8

	2.2 Automotive Function and Devices	9
	2.3 The Windows Xp Embedded Operating System	10
	2.4 Speech Recognition	13
	2.4.1 General Architecture for Controlling Devices	16
	2.4.2 Controlling Automotive Function using Speech Recognition General Approach	17
	2.5 Statistical Learning Framework	18
	2.6 Support Vector Machine (SVM)	20
	2.6.1 SVM Introductory Overview	21
	2.6.2 SVM Classifier	23
	2.7 GGobi 2D Function	25
CHAPTER III	METHODOLOGY	
	3.1 Introduction	26
	3.2 Research Approach	26
	3.3 Research Methodology	28
	3.4 General Workflow of SVM-DSW	34
	3.5 Analyze Requirement	35
	3.5.1 Software Requirement	35
	3.5.2 Hardware Requirement	35
	3.6 Summary	36
CHAPTER IV	DATA & DISCUSSION	
	4.1 Introduction	37
	4.2 Data Source	37
	4.2.1 Training and Testing Data	38
	4.3 Result	42
	4.4 Demo	49
	4.5 Summary	57

CHAPTER V	CONCLUSION	
	5.1 Introduction	58
	5.2 Advantages	59
	5.2.1 Commercialization	60
	5.2.2 Potential Beneficiaries	61
	5.3 Summary	61
	REFERENCES	63
	APPENDICES	70

LIST OF FIGURES

NO	FIGURES	PAGES
2.1	Human speech recognition process	14
2.2	General Speech Recognition Architecture	15
2.3	Illustrates the proposed system	17
2.4	Example object, RED and GREEN plots	22
2.5	Classic example of a linear classifier	22
2.6	Basic idea behind SVM	23
3.1	Research approach for SVM-DSW prototype development	27
3.2	MFCC Feature Extraction Procedure	28
3.3	Partial recognition simulation.	30
3.4	Embedded grammar rule.	31
3.5	DSW pseudo-code segment (prob = probability, curr = current, prev = previous, inc = increase, dec = decrease, recog = recognized/recognition, cnt = count).	32
3.6	Process workflow overview.	34
4.1	Performance comparison of the data augmentation technique.	41
4.2	Data visualization using GGobi for a) letter <i>b</i> and b) word <i>sorot</i>	48
4.3	Controlling Automotive Functions using Speech Recognition GUI	49
4.4	Recording voice using Matlab interface for “lampu luar buka”.	50
4.5	Wav file for “lampu luar buka”	51

4.6	Device info for “lampu luar buka”	52
4.7	Simulation for “lampu luar buka”	53
4.8	Recording voice using Matlab interface for “lampu luar tutup”.	54
4.9	Wav file for “lampu luar tutup”	55
4.10	Device info for “lampu luar buka”	56
4.11	Simulation after “lampu luar tutup” recognition process	57

LIST OF TABLES

NO	TABLE	PAGE
1.1	Word for control and devices	5
1.2	Example lists of commands	6
2.1	List of devices in a car for voice control	9
4.1	HTK workflow process for whole-word unit.	39
4.2	Confusion matrix and result analysis for a) baseline and b) proposed techniques.	44
4.3	Recognition performance benchmark	48

LIST OF ABBREVIATIONS

A2D	-	Analog to Digital
ASR	-	Automatic Speech Recognition
ANN	-	Artificial Neural Network
CAN	-	Controller Area Network
DBP	-	Dewan Bahasa Pustaka
DCT	-	Discrete Cosine Transform
DSW	-	Dynamic Shifting Window
FIR	-	Finite Impulse Response
GUI	-	Graphical user interface
HMM	-	Hidden Markov Model
HTK	-	Hidden Markov Toolkit
MDI	-	multiple document interface
MFCC	-	Mel-Frequency Cepstral Coefficients
OS	-	Operating System
PDA	-	Personal Digital Assistant
PROTON	-	Perusahaan Otomobil Nasional
QP	-	Quadratic Programming
SFS	-	Speech Filing System
SRM	-	Structural Risk Minimization
<i>SVA SDK</i>	-	<i>Sensory's VoiceActivation™ SDK</i>
SVM	-	Support Vector Machines

SVM-DSW	-	Support Vector Machine-Dynamic Shifting Windows
UTM	-	Universiti Teknologi Malaysia
VLSI	-	Very Large Scale Integration
Wifi	-	Wireless Fidelity

LIST OF APPENDICE

APPENDIX	TITLE	PAGE
A	List of commands	71
B	General Architecture for Speech Recognition in Microcontroller	74
C	Flow diagram chart for traditional speech recognition system	75
E	GUI for SVM-DSW	76
F	Recognition result for each iteration	78
G	HTK vs. DSW recognition benchmark	80

CHAPTER I

PROJECT OVERVIEW

1.1 Introduction

Speech recognition research has led to many commercial products this day. It is now very common to see devices being controlled by speech recognition. Smart home, voice dial and speech recognition computer software are many examples of its applications. Even with a lot of successful applications to this technology, sadly it is speech recognition with only English input. Although there are speech engines that recognize Malay words, there are almost no commercial products that are making the highlight of using speech recognition in Malay.

Statistically every year, there is a pattern of the increasing road accident in Malaysia. This is much to do with the attitude of the driver. Occasionally, we can see the driver is using the hand phone with one hand holding the steering wheel and the other to their hand phone. These actions lead to less concentration in driving and contribute to lack the response to the road condition. Things can go wrong in split second whereas the driver that cannot respond fast enough has to pay the price. There are also sometimes the driver is juggling with the car device such as searching a radio channel, gear changing or

just try to turn on the internal cabin light. The idea is that is it impossible to give a full concentration when the driver has to accomplish two things in the same time.

By combining two different backgrounds, a new challenge of resolving this situation can be done by developed a prototype of speech recognition system that can controlled car devices such as door windows, audio player or bonnet.

1.2 Background

The purposed of the design and development of a system for controlling automotive function using speech recognition research development is to allow consumer's car to control such device in their car by using their voice either to turn off or turn on devices and etc. To recognize consumer voice, SVM-DSW has been developed to achieve that purposed. SVM-DSW speech recognition engine will be embedded in a device which has a microcontroller to execute the processes. This device will be embedded in a car. So that, the consumer car will interfacing with this device to control such windows, horn, radio, lamp and etc in the car.

SVM-DSW speech recognition engine will be developed based on SVM method, developed using Matlab tools. In the end of the research development, the result will be compare between result from SVM-DSW and conventional method HMM.

1.3 Problem Statement

SVW-DSW speech recognition engine has been developed to give another ways to car's consumer to control their own device in a car by using a voice. So that, they can use a voice to turn off the radio, turn on the radio, enabled the horn and etc. However, to develop SVM-DSW speech recognition engine functional in a car, there are some problem that should be settle first. Some problems have been describes in general below:

- i. The tedious work of handpicking candidate speakers: filtering abnormal samples, data pre-processing: manual segmentation and feature extraction, and post-processing runs: classification and recognition; for all 16 speakers (four males and four females for each training and testing modules) had consume the entire semester.
- ii. Considering the times spend, we feel that the scope had to be cut-down. Instead of using a 69 sentences vocabulary, we opt to derive only 16 utterances from the same corpus. The focus is on how to exploit the Dynamic Shifting Window (DSW) potentials for speaker-independent continuous speech recognition task.
- iii. As it was stated in our objectives, the algorithm (DSW) will be tested against conventional approach (HMM) for benchmarking purposes and so have we successfully achieved the results needed using HTK. The expected outcome for DSW will come through shortly before the semester ends.
- iv. Despite the availability of segmentation residues from the recognition process, we do not wish to make an extensive comparison with the ones gained from DSW. Only a few examples between the two will be cross-examined for the sake of discussion, as the technique is never meant for segmentation.
- v. In order to conjure unbiased hypotheses from the experiment, the same recognition unit was used for both methods, namely the whole-word unit. However, we have also applied phone model for HMM to observe the differences. We argue that this model is quite ambiguous largely because it depends on trivial phonetic transcription, as there is no existing standard Malay phonetics for our vocabulary.
- vi. Foreseeing future advancements would be the detailed analytical analysis for the recognition accuracy of both techniques. We will discuss further about the pros and

contras for each system sub-components with hope in finding a better conclusion to which our scheme can be improved.

1.4 Aim

The aim of this research is to design and develop a system for controlling automotive function by using consumer's car voice to control a device in the car such as windows, radio, horn and etc which input by voice in Malay language.

1.5 Project Objectives

The purpose of this research is to design and develop a system for controlling automotive function using speech recognition. Consumer appliance will use their voice to control a device such as radio, windows, horn and etc. There are several objectives for this research. There are:

- i. To design and develop the proposed speech recognition system (for Malay) and the speech database based on an automotive control corpus
- ii. To implement, test and verify the proposed system using a prototype system capable of providing control for the functions of radio tuning, cruise control assistance, wind-screen wipers and air-conditioning control

1.6 Scope Project

Speech engine would be recognized word form the table 1.1. There are 49 words combining 9 words for devices and 36 words for control and lead to 69 types of commands sentences. The database consists of voices of 10 persons with 5 males and 5 females recording with continuous command sentences. The sentences is then break up as

a word after going a segmentation process. Total words for database are 450 words. The speech engine then recognizes the commands sentences word by word.

The whole system will be running in an embedded environment. After much considering, the Microsoft Windows XP Embedded is chose as the operating system for this project. The operating system has a very small image build up that can cater minimum requirement to run the speech recognition system. Prototype for this system can be described as a speech to text system.

Table 1.1: Word for control and devices

Word	Abbreviation	Word	Abbreviation
Air	AR	Letak	LE
Bahaya	BH	Lima	LI
Belakang	BG	Luar	LU
Berhenti	BT	Main	MA
Bonet	BN	Neutral	NE
Buka	BK	Pandu	PA
CD	CD	Pendingin Hawa	PD
Dalam	DM	Pengilap	PP
Depan	DE	Perlahan	PH
Dua	DU	Pintu	PT
Empat	EM	Radio	RD
Enam	EN	Rendah	RE
FM	FM	Satu	SA
Gear	GR	Sebelum	SM
Hon	HO	Selepas	SS
Isyarat	IS	Sembilan	SE
Kabus	KB	Sorot	SR
Kanan	KN	Suara	SU
Keluar	KL	Tiga	TI
Kipas	KP	Tinggi	TG
Kiri	KR	Tingkap	TK
Kosong	KS	Tujuh	TU
Kuat	KT	Tutup	TP
Lampu	LP	Undur	UN
Lapan	LA		

Table 1.2: Example lists of commands

No.	List of Commands			
1	Lampu	Luar	Buka	
2	Lampu	Luar	Tutup	
3	Lampu	Dalam	Buka	
4	Lampu	Dalam	Tutup	
5	Lampu	Isyarat	Kiri	Buka
6	Lampu	Isyarat	Kiri	Tutup
7	Lampu	Isyarat	Kanan	Buka
8	Lampu	Isyarat	Kanan	Tutup
9	Lampu	Bahaya	Buka	
10	Lampu	Bahaya	Tutup	

1.7 Thesis Arrangement

i. Project Overview.

First chapter that should be done is Project Overview. This chapter describes project overview: A Design and Development of a system for Controlling Automotive Functions using Speech Recognition. It contains of general overview research project: Speech Recognition, Automotive Control and our research target and also project objectives.

ii. Literature Review

After project overview has been defined, our project's problem definitions and literature review will be done. After problem definition and literature review done, the suitable methodology for this research project will be describes further in chapter III.

iii. Methodology

In this chapter, it will describe about the methodology, method and techniques that will be use in this system development life cycle. This chapter will be divided into two parts: project development methodology and system prototype development methodology.

iv. Project Design and Implementation

In this chapter, it consists of two main process developments: automotive controlling system development and speech recognition engine development. This chapter will describe a model to develop the system and the methodology based on SVM-DSW development process.

v. Result and Conclusion

This chapter will describe our result that we get from our system, speech recognition engine and word translation. After that, we will make a conclusion about the whole process development that we have done.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Controlling a devices or communicate with the computer by using a voice is one of the most effort that speech researcher want to achieve since a decades ago. One of the purposed of this Design and development of a system for controlling automotive function using speech recognition or SVM-DSW research is to contribute with that controlling device instead of improving speech recognition technique in controlling devices applications. Speech recognition for automotive control or better known as voice activated control is gaining acceptance in commercial automotive industry. Voice activated control for climate control, audio systems and telephone is already an important feature in automobiles such as Jaguars and Fords. The need to incorporate speech recognition into an automotive environment is already a common criterion because it promotes safe driving, improves automotive controls and also provides for more marketable automobiles.

The speech recognition is developed by integrating technology from *Sensory's VoiceActivation™ SDK (SVA SDK)* combining database with Support Vector Machine.

Sensory's VoiceActivation™ SDK (SVA SDK) is a comprehensive set of software components, tools, and methods that can be embedded Sensory's versatile, small footprint voice recognition technologies into a product. It provides a powerful *API* offering tremendous flexibility to accommodate the designed system architecture and product objectives and it is operating system independent.

2.2 Automotive Function and Devices

The purposed of SVM-DSW development was to control such devices in a car like horn, radio, window and etc by using consumer's car voice. In speech recognition terms, SVM-DSW used continuous vocabulary and small database. The input of this prototype is voice in spoken Malay. There are 69 command sentences consists of 9 control devices. Appendix A shows the full list of the devices and commands.

Table 2.1: List of devices in a car for voice control

No	Device
1	Air-Cond
2	Bonnet
3	Door
4	Gear
5	Horn
6	Lamp
7	Radio
8	Window
9	Wiper

From table 2.1 shows the list of the devices in a car that can be control by voice using this prototype SVM-DSW. There are several common commands to control those devices such turn off and turn on. Beside that, the prototype also can control the devices to open the door, close the door, change the gear and etc.

2.3 The Windows Xp Embedded Operating System

Windows XP Embedded is the embedded operating system that delivers the power of Windows in componentized form to allow developers to rapidly build reliable and advanced embedded devices. Based on the same binaries as Windows XP Professional, Windows XP Embedded contains over 10,000 individual feature components so developers can choose and achieve optimum functionality while managing or reducing footprint in a customized device image. Popular device categories for building operating systems using Windows XP Embedded include retail point-of-sale terminals, thin clients and advanced set-top boxes.

Windows XP Embedded delivers industry-leading reliability, security, and performance features and enhancements. The operating system software also provides the latest multimedia and Web browsing capabilities and contains extensive device support. In addition, Windows XP Embedded incorporates the latest embedded-enabling capabilities, such as support for multiple boot, storage, deployment and management technologies.

Based on the Win32 programming model, Windows XP Embedded reduce time-to-market by using familiar development tools such as Visual Studio .NET, working with commodity PC hardware and seamlessly integrating desktop applications

Microsoft XP Embedded image is created by Microsoft Windows Embedded Studio Target Designer. Target Designer is used to assemble a configuration to build into a run-time image for the target device. It accesses a component database that can be selected the components that can be add to a configuration. The component database contains the entire set of components included in the Windows XP Embedded operating system (OS).

Three methods can be chosen to create initial configuration:

- Use Target Analyzer to create a .pmq file that records the specific hardware, and then import the .pmq file into Target Designer as a configuration.

- Use Target Analyzer to create a .pmq file that records the specific hardware, and then import the .pmq file into Component Designer as a macro component. Macro component then can be add to the database and bring it into a configuration in Target Designer.
- Use one of the design templates provided with Windows Embedded Studio, and then add or remove components as necessary to suit specific application.

After building a run-time image for the target device, the deployment tools is used to prepare the target media, deploy the run-time image, and boot the target device.

The development process for creating and deploying run-time images consists of four major steps. Each major step must be completed in order:

- i. Create a new configuration or work with an existing configuration.

With a new configuration, components must be added. An existing configuration may already contain some components that can be edited Components also can be add to the existing configuration.

The Windows Embedded Studio tools include Target Analyzer, which can be use to collect information about the hardware in the target device, and then generate a configuration based on that information.

- ii. Add components to the configuration if necessary.

The components adding to a configuration determine the functionality of the run-time image that will be building. The process of selecting components can be divided into two parts: hardware and software. The category groups used to organize components support this approach. At the root of the category tree are two groups: Hardware and Software. If Target Analyzer tool is used to create the configuration, most, if not all, of the hardware components required for the hardware will already be selected. This leaves with only the software component to select.

If no configuration is currently open, these options are unavailable. Also, if the component cannot be added to a configuration for any other reason, the Add option is unavailable and the components cannot be dragged into the configuration editor.

The component database can have multiple versions of the same component. If more than one version exists in the database, you specify the version to add by choosing the Add Version menu item. The most recent version is marked with (current) and appears at the top of the menu.

When a component is added to a configuration, the configuration editor scrolls down to display the component that was just added. However, the current tree node selection does not change in the configuration editor.

- iii. Dependencies checking for all the components in the configuration.

A dependency is a functional relationship between two or more components. Before building the configuration into a run-time image, a dependency check should be run on the configuration to ensure that all the component dependencies have been resolved.

Target Designer can be set to automatically resolve certain component dependencies. If the dependency check encounters components with unresolved dependencies, a task describing the requirement is added to the task list for each unresolved dependency.

Each task in the Tasks list is actually a filter that displays a dialog box listing only those components that can resolve the corresponding dependency.

2.4 Speech Recognition

Speech recognition is an alternative to traditional methods of interacting with a computer, such as textual input through a keyboard. An effective system can replace, or reduce the reliability on, standard keyboard and mouse input. This can especially assist the following:

- People who have little keyboard skills or experience, who are slow typist, or do not have the time or resources to develop keyboard skills.
- Dyslexic people, or others who have problems with character or word use and manipulation in a textual form.
- People with physical disabilities that affect either their data entry, or ability to read (and therefore check) what they have entered.

Speech and understanding voice message by human is a complex process. Factors like height, weight, sex, teeth and lips can give an impact to their speech. Voice processing by human can be simplified as below.

Processing sequence in the human auditory system.

- Fixed filter which represents the transfer from free field to eardrum and through the middle ear.
- A bank of logarithmically spread bandpass filters in cochlea.
- Dynamic compression, when mechanical energy is transformed to neural signals by the hair cells.
- Periodicity estimation at each band.
- Integration of the band-wise processed signals and further calculations. This takes place in the central nervous system (brains).

Human perception of speech starts with receiving signal by the ear. It will then pass the membrane basilar in the inner ear where the signal will be analyzed. The analyzed signal will pass to neural transducer that convert the signal into activity signal

on the auditory nerve and the brain will translate and understood the speech. Figure 2.1 show the scenario of human speech recognition process.

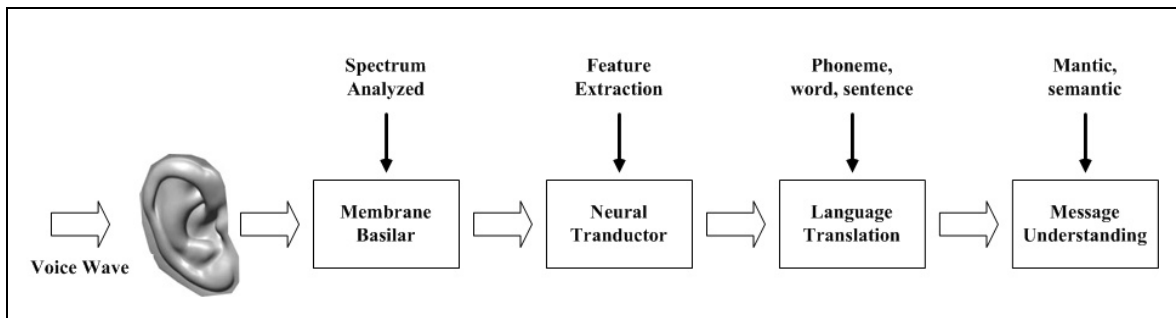


Figure 2.1: Human speech recognition process

A speech recognition system consists of the following:

- A microphone, for the person to speak into.
- Speech recognition software.
- A computer to take and interpret the speech.
- A good quality soundcard for input and/or output.

At the heart of the software is the translation part. Most speech recognition software breaks down the spoken words into phonemes, the basic sounds from which syllables and words are built up. These are analyzed to see which string of these unit best “fits” an acceptable phoneme string or structure that the software can derive from its dictionary.

It is a common misassumption that such a system can just be used “out of the box” for work purposes. The system has to train to recognize factors associated with the user’s voice, for examples speed, pitch. Even after this training, the user often has to speak in a clear and partially modified manner in order for his or her spoken words to be both recognized and correctly translated.

Most speech recognition software is configured or designed to be used on a stand-alone computer. However, it is possible to configure some software in order to be used over a network. We can classify speech recognition tasks and systems along a set of dimensions that produce various tradeoffs in applicability and robustness. A speech recognition system can be used in many different modes (speaker dependent or independent, isolated / continuous speech, for small or large vocabulary). Figure 2.2 show the general speech recognition architecture which it contains two main components, Features Extraction and Speech Recognizer. This architecture received speech voice as an input and text as an output.

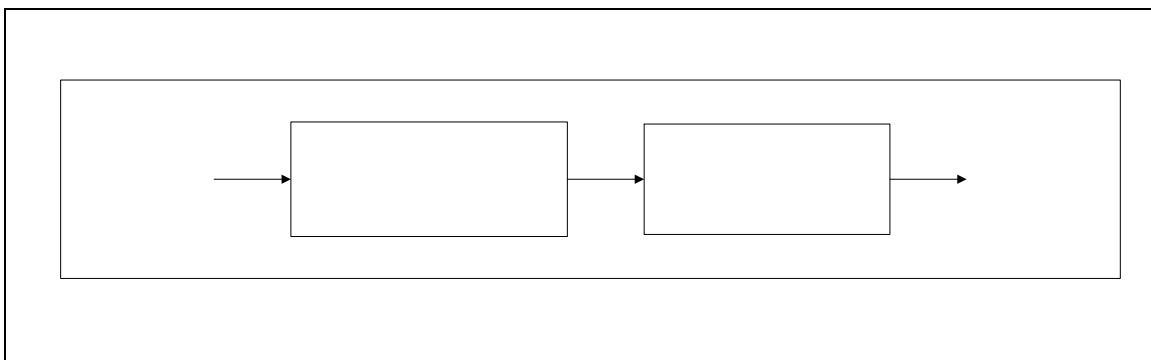


Figure 2.2: General Speech Recognition Architecture

Isolated word versus continuous speech: Some speech systems only need identify single words at a time (e.g., speaking a number to route a phone call to a company to the appropriate person), while others must recognize sequences of words at a time. The isolated word systems are, not surprisingly, easier to construct and can be quite robust as they have a complete set of patterns for the possible inputs. Continuous word systems cannot have complete representations of all possible inputs, but must assemble patterns of smaller speech events (e.g., words) into larger sequences (e.g., sentences).

Speaker dependent versus speaker independent systems: A speaker dependent system is a system where the speech patterns are constructed (or adapted) to a single speaker.

Speaker independent systems must handle a wide range of speakers. Speaker dependent systems are more accurate, but the training is not feasible in many applications. For instance, an automated telephone operator system must handle any person that calls in, and cannot ask the person to go through a training phase before using the system. With a dictation system on your personal computer, on the other hand, it is feasible to ask the user to perform a hour or so of training in order to build a recognition model.

Small versus vocabulary systems: Small vocabulary systems are typically less than 100 words (e.g., a speech interface for long distance dialing), and it is possible to get quite accurate recognition for a wide range of users. Large vocabulary systems (e.g., say 20,000 words or greater), typically need to be speaker dependent to get good accuracy (at least for systems that recognize in real time). Finally, there are mid-size systems, on the order to 1000-3000 words, which are typical sizes for current research-based spoken dialogue systems.

Some applications can make every restrictive assumption possible. For instance, voice dialing on cell phones has a small vocabulary (less than 100 names), is speaker dependent (the user says every word that needs to be recognized a couple of times to train it), and isolated word. On the other extreme, there are research systems that attempt to transcribe recordings of meetings among several people. These must handle speaker independent, continuous speech, with large vocabularies. At present, the best research systems cannot achieve much better than a 50% recognition rate, even with fairly high quality recordings.

2.4.1 General Architecture for Controlling Devices

The proposed system comprises four main components; i.e. input speech acquisition, input feature extraction, acoustic modeling and word matching. Analogue speech signal is fed into the system through the use of a microphone. This analogue signal is then converted to a digitized form. The signal is then preprocessed to extract

usable features for the acoustic modeling phase. In this phase the input speech will be matched word for word through a statistical model for the need of recognizing the correct voice command.

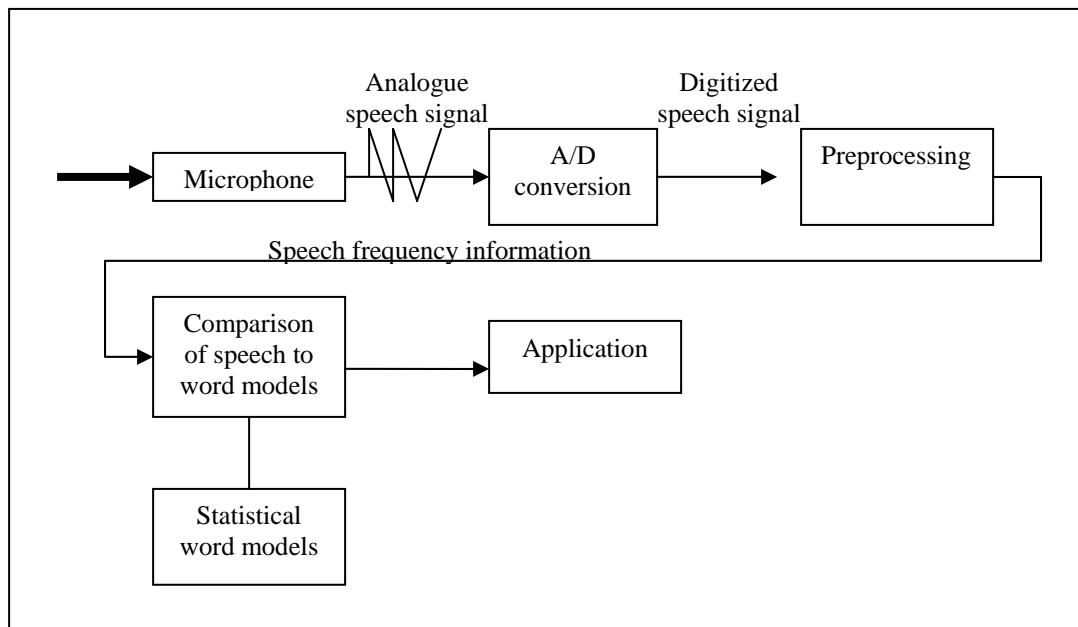


Figure 2.3: Illustrates the proposed system

2.4.2 Controlling Automotive Function using Speech Recognition General Approach

There are several stage continuously perform stage by stage in purposed to achieved automotive controlling device using speech recognition. All stages lists a below:

- i. Raw speech input is converted from analog to digital (A2D) waveform via a uni-directional microphone and consequently stored in WAVE sound file format.
- ii. Waveform graph representing the digitized speech signal is automatically displayed upon completion of each recording session. The waveform consisting signed integer values contained in the WAVE data chunk (main body of the WAVE file).

- iii. Further characteristic of the WAVE file such as sampling rate (8000, 11025, 22050), channel type (mono/stereo) and bits per sample (8/16 bits) are also being extracted from the WAVE file header. This information is much needed both in the early and later stages.
- iv. Based on the plotted waveform, a typical start-endpoint detection algorithm (short-time energy and zero-crossing rate) is applied to locate the actual speech activities from the entire region. Dismissing the beginning and trailing silence part of speech will reduce the processing load. The result, a connected string of words, however, is still lengthy and inappropriately oversized. Thus an additional or modified endpoint algorithm is used to spot the word boundaries and isolates each word.
- v. Speaking in graphical user interface (GUI) context, the program will be able to open an array of displays comprises a plotted waveform before and after affects of the endpoint detection, a frequency response graph before and after applying the Finite Impulse Response (FIR) filters, LPC or cepstral coefficients and the resulting feature vector values. By using multiple document interface (MDI) feature in Windows programming, more than one speech sample can be open simultaneously. This supports the need for comparison.

2.5 Statistical Learning Framework

Neural networks have also been applied to speech recognition owing to several advantages they offer over the typical HMM systems. Neural networks can learn very complex non-linear decision surfaces effectively and in a discriminative fashion. However, their estimation process is significantly more computationally expensive than HMMs and they are typically formulated as classifiers of static data. This has led to the development of several connectionist approaches where the neural networks are

embedded in a HMM framework. The performance of these hybrid systems have been competitive with many HMM-based systems and typically require a significantly reduced parameter count. The hybrid connectionist systems also provide a way to mitigate some of the assumptions made in HMM systems that we know are incorrect for the human speech process. One such significant assumption is that of independence of observations across frames. Hybrid systems mitigate this problem by allowing the neural network classifiers to classify based on several frames of acoustic data at a time. A similar approach will be pursued in this dissertation by processing multi-frame data.

Although the final system configurations for connectionist systems are simpler than HMM based systems, their use has been limited because of various limitations listed below:

- **Generalization:** Neural networks have been known to overfit data unless specific measures are taken to avoid that. These measures typically include some form of cross-validation which can be restrictive when the amount of training data is limited to start with.
- **Optimization Process:** Neural network learning is based on the principle of empirical risk minimization via the back-propagation algorithm. Though this guarantees good performance on the training data, obtaining a bound on the performance on the test data is not easy.
- **Model Topology:** In most connectionist hybrid systems the topology of the neural network classifiers needs to be fixed prior to the estimation process. This is not always easy without expert knowledge of the data. Techniques do however exist to learn connections automatically but are expensive.
- **Convergence:** Convergence of the optimization process has been the biggest drawback of neural networks. Convergence is typically an order of magnitude slower than ML estimation of HMM parameters. Both ML estimation using the EM algorithm and estimation of parameters of the neural networks do not guarantee reaching a global maximum unless measures are taken to perturb the system from time to time which increases the possibility of reaching the global maximum.

The need for discrimination and classifiers with good generalization and convergence properties that can be used for speech recognition has led us to look at a new machine learning paradigm called the support vector machines (SVM) which forms the basis of this dissertation.

2.6 Support Vector Machine (SVM)

Recognition process is a challenges task. It required a lot of effort and computational process to recognize a single word from human voice. There are many techniques and method that provided to recognize human speech and also depend on what kind of purpose the recognition process must be done.

Hidden Markov Models (HMMs) are, undoubtedly, the most employed core technique for Automatic Speech Recognition (ASR). During the last decades, research in HMMs for ASR has brought about significant advances and, consequently, the HMMs are currently accurately tuned for this application. Nevertheless, we are still far from achieving high-performance ASR systems. Some alternative approaches, most of them based on Artificial Neural Networks (ANNs), were proposed during the last decade. Some of them tackled the ASR problem using predictive ANNs, while others proposed hybrid (HMM-ANN). However, despite some achievements, none of these approaches could outperform the results obtained with HMMs and, nowadays, the preponderance of Markov Models is a fact. In the last decade, however, a new tool appeared in the field of machine learning that have proved its capability to overcome many of the problems of techniques as ANNs. The Support Vector Machines (SVMs) are effective discriminant classifiers capable of maximizing the error margin. As opposed to ANNs, they have the advantage of being capable to deal with samples of a very higher dimensionality. Also, their convergence to the minimum of the associated cost function is guaranteed as a simple problem of quadratic programming (QP). Besides, instead of only minimizing the

empirical risk, they also try to minimize the “structural risk”, being the solution a compromise between empirical error and generalization capability.

These characteristics have made SVMs very popular and successful in many fields of application. Nevertheless, in order to use them in a problem of speech recognition, some limitations must be overcome. One of them is the number of training samples they can deal with that, in spite of the apparition of techniques as Sparse SVM, is still limited to a few thousands. Another problem of SVMs is that, in their original formulation, they are restricted to work with input vectors of fixed dimension (although nowadays there are some solutions to cope with this problem, as we will see). Finally, another limitation is that SVMs *only* classify, but they don't give us a reliable measure of the probability of the correctness of the classification. This can cause problems in recognition, where without a concrete value of probability we can't carry out some algorithms as Viterbi, to look for the most probable sequence of recognition units.

2.6.1 SVM Introductory Overview

SVM are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in the illustration below. In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the right side of which all objects are GREEN and to the left of which all objects are RED. Any new object (white circle) falling to the right is labeled, i.e., classified as GREEN (or classified as RED should it fall to the left of the separating line).

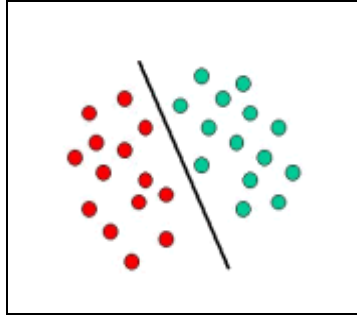


Figure 2.4: Example object, RED and GREEN plots

The figure above is a classic example of a linear classifier, i.e., a classifier that separates a set of objects into their respective groups (GREEN and RED in this case) with a line. Most classification tasks, however are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases). This situation is depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line). Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. SVM are particularly suited to handle such tasks.

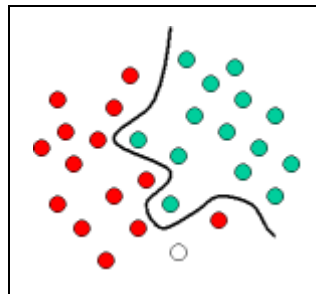


Figure 2.5: Classic example of a linear classifier

The illustration below shows the basic idea behind SVM. Here we see the original objects (left side of the schematic) mapped, i.e., rearranged, using a set of mathematical functions, known as kernels. The process of rearranging the objects is known as mapping (transformation). Note that in this new setting, the mapped objects (right side of the

schematic) is linearly separable and, thus, instead of constructing the complex curve (left semantic), all we have to do is to find an optimal line that can separate the GREEN and the RED objects.

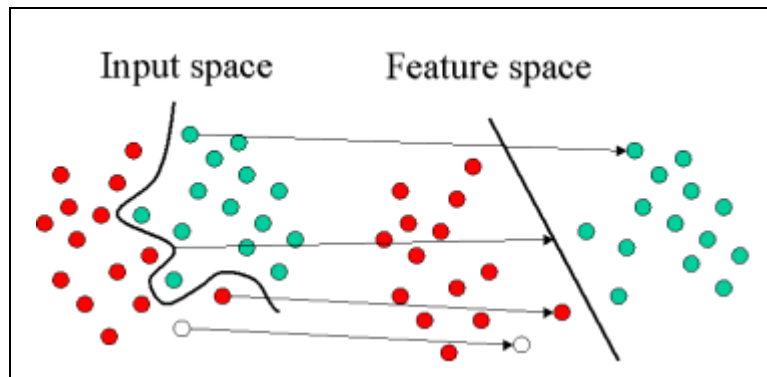


Figure 2.6: Basic idea behind SVM

2.6.2 SVM Classifier

Some of the generalization properties of neural networks have been mentioned in the previous section. Why is generalization important? HMM-based speech recognition systems perform very well on closed-loop tests but performance degrades significantly on open-loop tests. The performance of systems on speaker-dependent tasks is significantly better than on speaker-independent tasks. This can be attributed to the fact that most systems do not generalize well. There is a definite need for systems with good generalization properties where the worst-case performance on a given test set can be bounded as part of the training process without having to actually test the system. With many real-world applications where open-loop testing is required, the significance of generalization is further amplified.

As mentioned in a previous section, empirical risk minimization is one of the most commonly used optimization criteria to estimate classifiers. However, there can be

several configurations of the classifier that can achieve minimum risk on the training set. This is one of the reasons why neural networks can get stuck in local saddle points. The problem then is to decide on the configuration that has the least upper bound on the expected test set error. This is the principle of structural risk minimization (SRM). Support vector machines are founded on this principle and the result of SRM is a classifier with the least expected risk on the test set and hence good generalization.

SVMs in their simplest form are hyperplane classifiers. The power of SVMs lies in their ability to implicitly transform data to a high dimensional space and to construct a linear binary classifier in this high dimensional space. Since this is done implicitly, without having to perform any computations in the high dimensional space, neither the dimensionality of the data nor the sparsity of data in the high-dimensional space is a problem with SVMs.

SVMs have been applied successfully on several kinds of classification problems and have consistently performed better than other non-linear classifiers like neural networks and mixtures of Gaussians. The dataset that propelled SVMs to prominence in the early 90's was the US Postal Service digit data on which the SVMs achieved the best numbers reported. The development of efficient optimization schemes led to the use of SVMs for classification of larger tasks like text-categorization.

There were some initial efforts to apply SVMs to speaker recognition in the early 90's. This effort had limited success because of the lack of efficient implementations of the SVM estimation process at that time. SVMs have also been applied to simple phone classification tasks and the results have been very encouraging. Notice however that all the above classification tasks have one common feature - these are all static classification tasks. SVMs are not designed to handle temporal structure of data. Speech however evolves with time and we need to address this problem in order to harness the advantages of SVMs for speech recognition. This is the primary contribution of this dissertation wherein we have developed a hybrid SVM/HMM framework with the HMM structure being used to handle the temporal evolution of speech and SVMs being used to

discriminatively classify frames of speech. The end result is a first successful application of SVMs to continuous speech recognition.

2.7 GGobi 2D Function

Based on the result from those experiments that been done, some software has been used in purposed to describe more detailed and easy understanding. One of the software that been used is GGobi. GGobi 2D is an interactive graphical software for exploratory data analysis. By using GGobi, it is easy to perform a simple task with virtually no instruction. What is needed just some cursory knowledge of the developments in interactive statistical graphics. In another word, GGobi is an open source visualization program for exploring high-dimensional data. It provides highly dynamic and interactive graphics such as tours, as well as familiar graphics such as the scatterplot, barchart and parallel coordinates plots. Plots are interactive and linked with brushing and identification.

By using GGobi, the results from the experiment that been done can be display in graphical view. It is one way to explain about the result compare to real situation in a car with real radio, window and etc. From the graphical views, it also can show the output from the recognition process. As been describes before in chapter I, this research development is about to control such devices in a car like radio, window, horn and etc. so that, by using GGobi, the result also can be translated in graphical view. It is to ensure that the algorithm in SVM-DSW in well perform with the input (consumer's voice).

CHAPTER III

METHODOLOGY

3.2 Introduction

Defining the project's methodology is an important task, as it can give guidelines about activities that need to be performed in order to successfully develop a system. Moreover it helps to achieve the project's objectives and vision, as well as solving the background problems. This chapter discusses the methodology of the research project: A Design and Development of system for Controlling Automotive Functions using Speech Recognition. This chapter will give a clear view on the methodology used by describing the framework of SVM-DSW. This chapter attempts to provide clear guidelines on how the project goal and objectives are accomplished.

3.2 Research Approach

This sub-chapter describes the design and development of a system for controlling automotive function using speech recognition research approach to achieve research

objectives. Generally, there are three steps in research approach for this research development (figure 3.1) and will be describes in the next section.

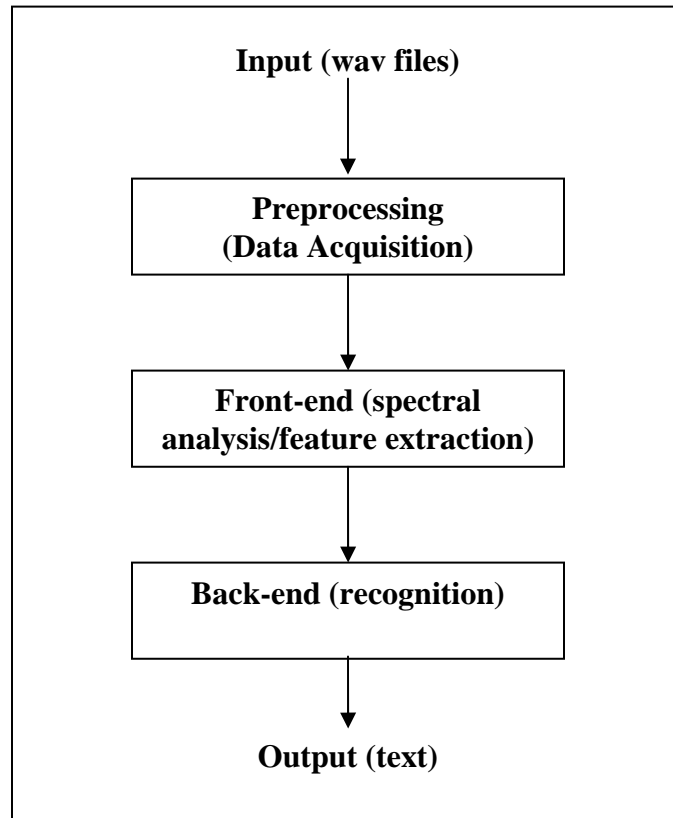


Figure 3.1: Research approach for SVM-DSW prototype development

i. Preprocessing (data acquisition)

Multi-speaker recording sessions with manual speech segmentation (hand labeling) for the training phase (half of the whole samples). There are 100 hour of *.wav files have been recorded for training process purposed. There are 50 speakers consists of 25 male and 25 female involved in recording procedure. Those speakers consist of different ethnic and age. For each wav files it contains 3 second per frame.

ii. Front-end (spectral analysis/feature extraction)

Mel-Frequency Cepstral Coefficients (MFCC) – conventional approach for robust speech feature extraction technique. MFCC is a signal processing techniques to extract acoustic features from the speech waveform. A diagram of the MFCC feature extraction procedure is shown in figure 3.2. First, the input waveform signal goes through a Mel-scaled filter bank. Then it is followed by low pass filtering and downsampling. Finally, discrete cosine transform (DCT) is performed on the log-energy of the filter outputs.

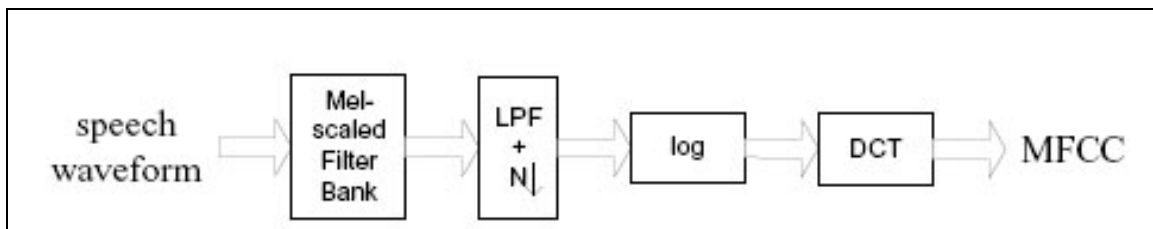


Figure 3.2: MFCC Feature Extraction Procedure

iii. Back-end (recognition)

Support Vector Machines-Dynamic Shifting Window (SVM-DSW) – a combination of the most recent and discriminate type of classifiers (SVM) with enhanced level building algorithm (DSW).

3.3 Research Methodology

The mechanism of DSW starts off by comparing the initial segment's probability score with a pre-determined threshold. If it exceeds, a smaller chunk of features will be extracted and appended to the original segment; indicating that the region of interest is nearby. Otherwise, larger segments will continue to progress along the feature-arrays. The procedure ultimately pin-points the exact location of the speech segment, that corresponds to a particular word. This is achieved by either increasing or decreasing the

number of features for the current segment accordingly. Apart from the probability score cues, DSW also governed by an embedded grammar rule; this limits the possibility of recognizing out-of-vocabulary utterances. In other words, it tries to maximize the probability of a known string of words belonging to an anonymous array of features. With respect to conventional grammar induced algorithm which calculates the probability of words being recognized beforehand, DSW embeds this grammar rule in real-time. Owing to SVM's confidence in recognition and translating it into probability estimation, we felt that there is no need for extra computation on the language counterpart itself. Instead of producing misleading conclusion when use in isolation (acoustic and grammar attributes), ours would drive the recognition process faster and more accurate (refer Figure 3.3, 3.4 and 3.5).

- vii. The recognition scheme, denote as Dynamic Shifting Window (DSW), is now robust against silence and noise factors that maybe existed along the path of speech signal as well as it can handles variability in the rate of speaking. Hence, we have omitted endpoint detection and frame normalization processes in the current methodology.
- viii. Despite the gains, it does have a few shortcomings. The size of window may vary from one speaker to another; thus, a speaker-dependent training mode could bare better results. In addition, segmentation error could occur in case of insufficient training sample.
- ix. Only two control variables play major role in the whole algorithm, namely the initial window size and the shifting increment. The first parameter determines the initial guess where each word in the utterance is located. The latter one positions (moves and resizes) the window to fit the entire word.
- x. Multi-speaker recognition has been tested and benchmarked against Hidden Markov Toolkit (HTK) with very promising outcomes. DSW which is based on Support

Vector Machines (SVM) classifier has excels in both segmentation and recognition accuracy.

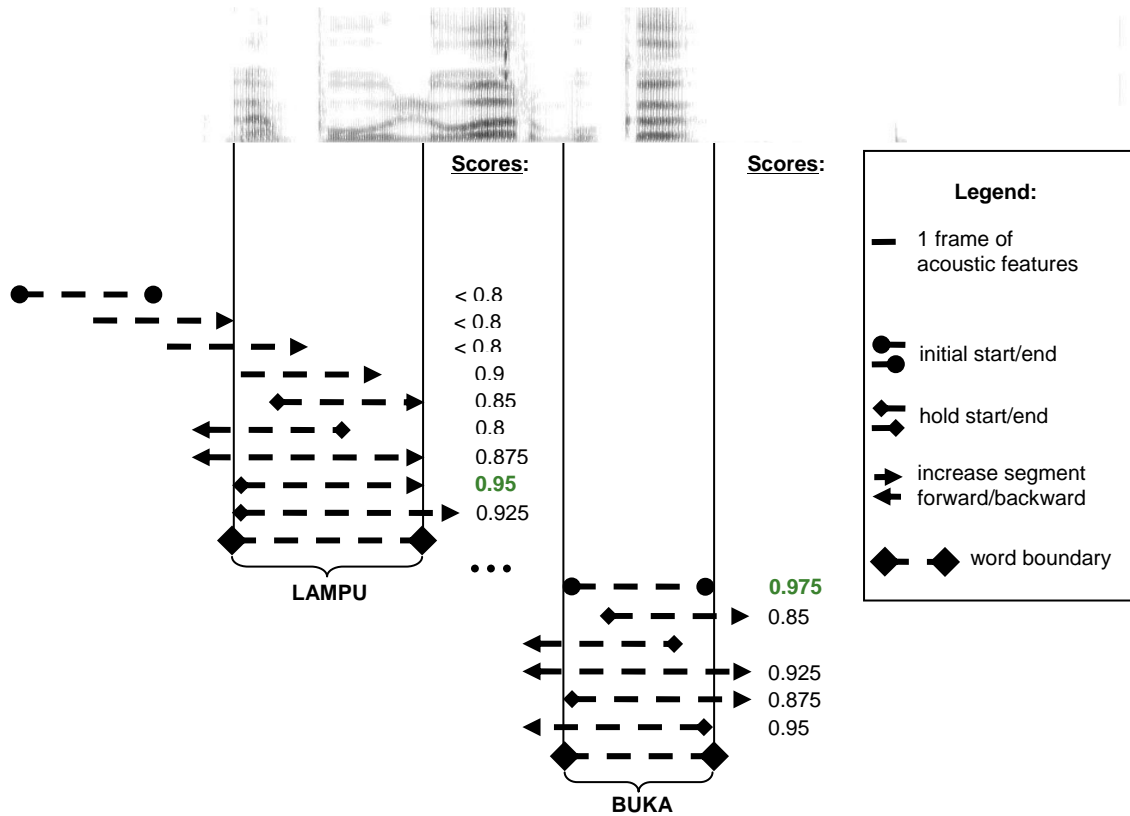


Figure 3.3: Partial recognition simulation.

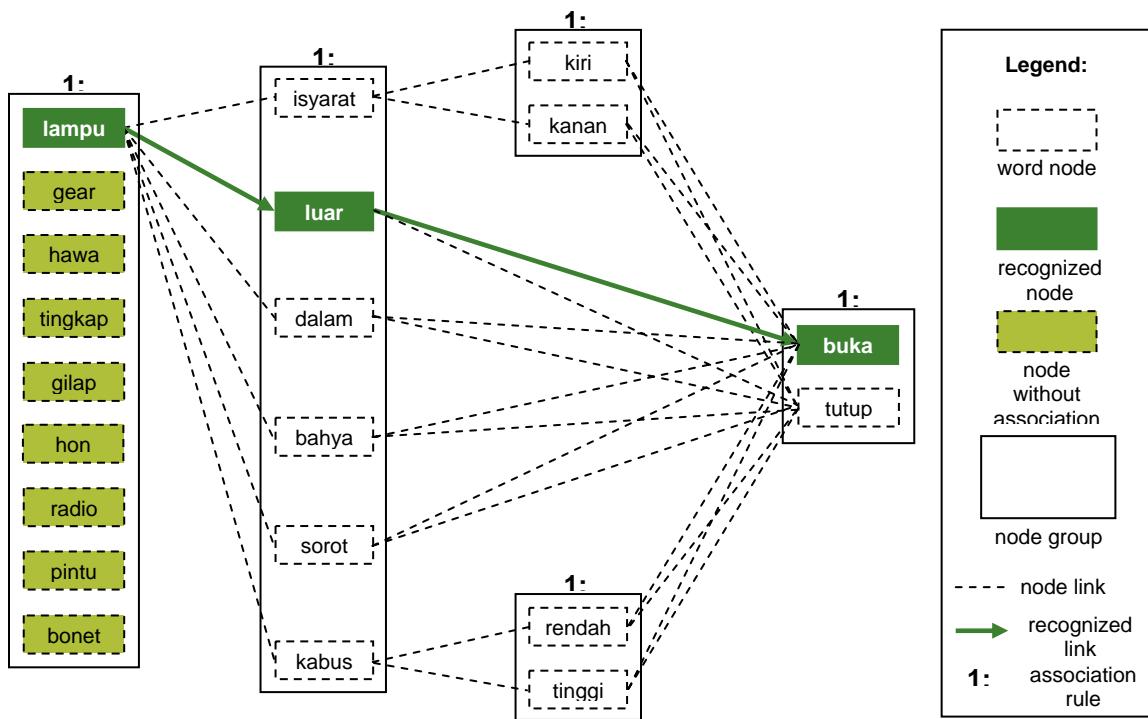


Figure 3.4: Embedded grammar rule.

If (curr_prob >= **Probability Threshold**)

If (curr_recog_word = null)

prev_prob = curr_prob;
prev_recog_word = curr_recog_word;

Else If (prev_recog_word = curr_recog_word)

unstable_recog_cnt = 0;

If (prev_prob < curr_prob)

prev_prob = curr_prob;
prev_recog_word = curr_recog_word;
prob_dec_cnt++;
prob_inc_cnt = 0;

Else If (prev_prob >= curr_prob)

prev_prob = curr_prob;
prev_recog_word = curr_recog_word;
prob_inc_cnt++;

If (prob_dec_cnt > 1) && (prob_inc_cnt > 3)

switch train model;
set new **Backtrack Size**;
set new **Segment Size**;
prev_prob = prev_recog_word = null;
prob_dec_cnt = prob_inc_cnt = unstable_recog_cnt =
0;

Else If (prev_recog_word <> curr_recog_word)

unstable_recog_cnt++;

```

If (prob_increase_count > 2)
    switch train model;
    set new Backtrack Size;
    set new Segment Size;
    prev_prob = prev_recog_word = null;
    prob_dec_cnt = prob_inc_cnt = unstable_recog_cnt = 0;

Else If (prob_dec_cnt > 3)
    switch train model;
    set new Backtrack Size;
    set new Segment Size;
    prev_prob = prev_recog_word = null;
    prob_dec_cnt = prob_inc_cnt = unstable_recog_cnt = 0;

Else If (unstable_recog_cnt > 1)
    prev_prob = prev_recog_word = null;
    prob_dec_cnt = prob_inc_cnt = unstable_recog_cnt = 0;
    prob_threshold_cnt = 0;

Else If (curr_prob < Probability Threshold)
    prob_threshold_cnt++;

If (prob_threshold_cnt > 1)
    prev_prob = prev_recog_word = null;
    prob_dec_cnt = prob_inc_cnt = unstable_recog_cnt = 0;

```

Figure 3.5: DSW pseudo-code segment (prob = probability, curr = current, prev = previous, inc = increase, dec = decrease, recog = recognized/recognition, cnt = count).

3.5 General Workflow of SVM-DSW

- i. The recognition scheme, denote as Dynamic Shifting Window (DSW), is now robust against silence and noise factors that maybe existed along the path of speech signal as well as it can handles variability in the rate of speaking. Hence, we have omitted endpoint detection and frame normalization processes in the current methodology.
- ii. Despite the gains, it does have a few shortcomings. The size of window may vary from one speaker to another; thus, a speaker-independent training mode could bare better results. In addition, segmentation error could occur in case of insufficient training sample.
- iii. Only two control variables play major role in the whole algorithm, namely the initial window size and the shifting increment. The first parameter determines the initial guess where each word in the utterance is located. The latter one positions (moves and resizes) the window to fit the entire word.

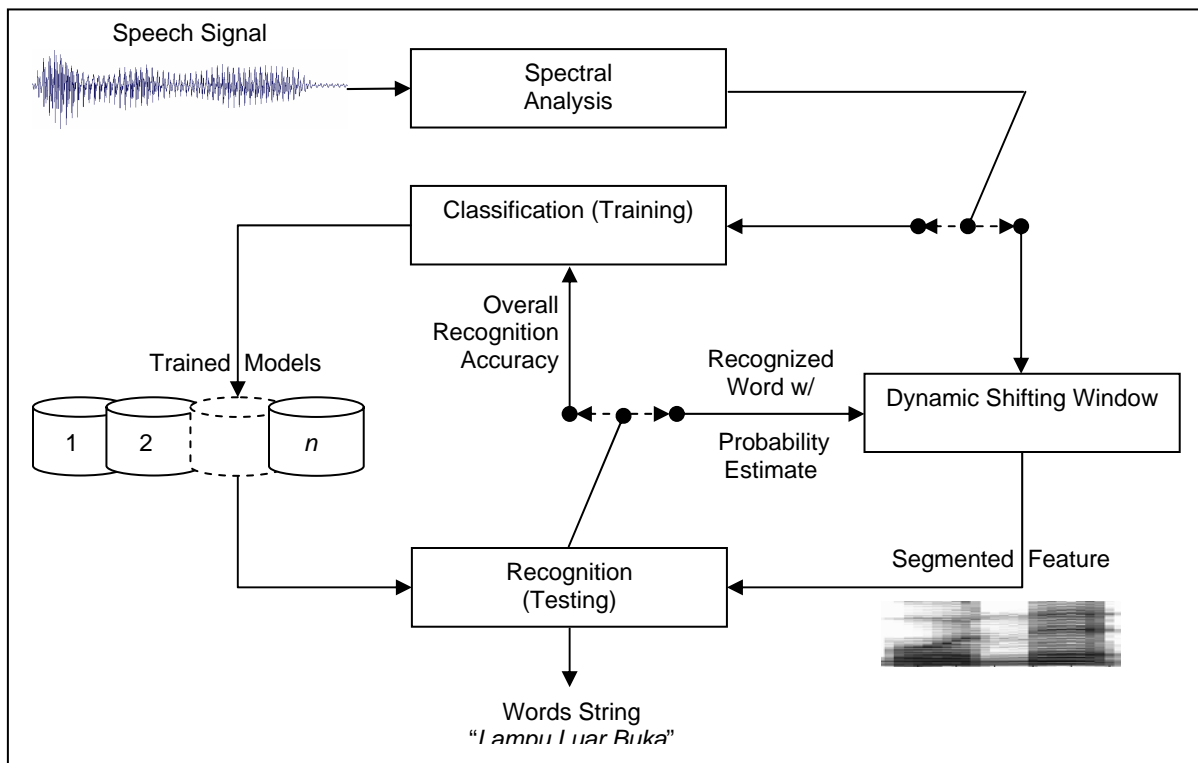


Figure 3.6: Process workflow overview.

3.5 Analyze Requirement

To develop a prototype of Intelligent Agent for Speech Recognition and Translation, this prototype needs two requirements, software and hardware to achieve the process development.

3.5.1 Software Requirement

There are several types of software involved in this research prototype.

i. Matlab

Matlab handles numerical calculations and high-quality graphics, provides a convenient interface to built-in state-of-the-art subroutine libraries, and incorporates a high-level programming language. For training process in speech recognition system, matlab has been used for training modules. Data that been collected will be process and it called a training data.

ii. C Programming Language

C programming language has been used to implement SVM technique for recognition process.

3.5.2 Hardware Requirement

Hardware also is a main component parts in prototype development. Generally, there is a computer applied in prototype development especially for speech recognition processes development (computational) and one personal computer for collecting data training and preprocessing (data acquisition) process. This hardware specification is as below:

i. Personal Computer (computational development)

- a. Pentium IV 1.8 GHz Processor

- b. 40 GB Hard Disk
 - c. 512 Mb RAM
 - d. Keyboard and Mouse
 - e. Monitor 17”
 - f. Sound Card
- ii. Personal Computer (preprocessing – data acquisition)
 - a. Pentium IV 1.6 GHz processor
 - b. 35 GB Hard Disk
 - c. 512 Mb RAM
 - d. Keyboard and Mouse
 - e. Monitor 17”
 - f. Sound Card and Microphone

3.6 Summary

A well designed and robust methodology is needed to run such an intensive research as SVM-DSW. The methodology designed and followed is shown to be the best fit for this research study. The next chapter will discuss the detail data and discussion about this prototype.

CHAPTER IV

DATA AND DISCUSSION

4.1 Introduction

This chapter describes about two topics, the data and discussion about the result that been achieved from the experiments based on SVM-DSR method. In this chapter also, the data for this research will be explain in detailed such as data format, how the data collected and processed and etc. The result form the experiments will be describe to show the performance and contribution that SVM-DSR offer to improve the performance of controlling automotive functions using voice.

4.2 Data Source

Collected speech samples have been down sampled to 8 kHz (telephone line condition) without affecting the recognition accuracy, thus stepping up the process by minimizing computation.

Speech Filing System (SFS) 4.6 was used for automatically extracts the continuous speech into discrete speech signals for training purposes. Do note that the word-segment hand-labeling process was done beforehand.

Corrections of mislabeled speech segments were then follows. Other errors constituted of voice clippings, cut-offs and mispronunciations. These types of errors usually were rectified by removing the whole sequence and reproducing new substitute.

4.2.1 Training and Testing Data

As opposed to HMM and other techniques, SVM utilizes only minimal feature (8th order Mel-Frequency Cepstral Coefficients instead of 12th order per frame) to classify discriminately. By using whole word as the recognition unit, much of the contextual information are retained, therefore we do not have to represent each frame deliberately. Frame normalization or time alignment, which implies a theoretical issue for static classifier such as SVM; appear to be practically unnecessary in this research. A simple zero padding at the end of each word's frame is sufficient to produce good output.

A new criterion for SVM's model selection has been formulated to suit the purpose of the project. Conventional model selection relies solely on classification accuracy. Our algorithm requires the use of probability estimate to perform classification. This is vital owing to the fact that the highest probability determines the recognized word. Thus, the model selection in SVM must be based upon classification accuracy that undertakes probability estimates. Subsequently, selected models will be used as 'matching templates' in the recognition paradigm.

Previous version of DSW did not allow both ends of the window segment to progress, which in turn contributes toward the rigidity of the algorithm. Flexibility are then endowed in the technique by simply adding adaptable begin and end points of the

window itself. This alleviates confusion of co-articulation effects among inter-words that have high dependency.

In order to conjure unbiased hypotheses from the experiment, the same recognition unit was used for both methods, namely the whole-word unit. However, we have also applied phone model for HMM to observe the differences. We argue that this model is quite ambiguous largely because it depends on trivial phonetic transcription, as there is no existing standard Malay phonetics for our vocabulary.

We managed to obtain two outcomes from HTK (apart from segmentation attributes): recognition based on whole-word unit and phone unit. The complexity of the recognition task can be calculated as 16 (sentences) x 10 (repetitions) x 8 (speakers) = 1280 sentences = 56 (words) x 10 (repetitions) x 8 (speakers) = 4480 words (on a 13 words vocabulary). HTK scores 76.72% and 95.23% sentence accuracies for the first (whole-word) and second (phone) procedures respectively. This justified the effectiveness of HMM in dealing with sub-word units. Our target however is to approach or exceed HTK's whole-word performance milestone and consequently extend the DSW capabilities into a phone-like unit based recognizer (refer **Table 4.1**).

Table 4.1: HTK workflow process for whole-word unit.

1) Building the task grammar (a "language model")	
\$model1 = KIRI KANAN;	
\$model2 = RENDAH TINGGI;	
\$model3 = LUAR DALAM BAHAYA SOROT ISYARAT \$model1 KABUS	
\$model2;	
\$model4 = BUKA TUTUP;	
(SILENCE LAMPU \$model3 \$model4 SILENCE)	
2) Constructing a dictionary for the models	
BAHAYA	bahaya sp
BUKA	buka
DALAM	dalam sp

ISYARAT	isyarat sp
KABUS	kabus sp
KANAN	kanan sp
KIRI	kiri sp
LAMPU	lampu sp
LUAR	luar sp
RENDAH	rendah sp
SILENCE []	sil
SOROT	sorot sp
TINGGI	tinggi sp
TUTUP	tutup
3) Creating transcription files for training data	
<pre> #!MLF!# "/M1SET01.lab" Lampu Luar Buka . "/M1SET02.lab" Lampu Luar Buka . (etc.) </pre>	
4) Encoding the data (feature processing)	
<pre> # Coding parameters TARGETKIND = MFCC_0_D_A TARGETRATE = 100000.0 SAVECOMPRESSED = T SAVEWITHCRC = T </pre>	

WINDOWSIZE = 250000.0

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 26

CEPLIFTER = 22

NUMCEPS = 12

5) (Re-)training the acoustic models

~o <VecSize> 39 <MFCC_0_D_A>

~h "Lampu"

<BeginHMM>

<NumStates> 5

<State> 2

<Mean> 39

0.0 0.0 0.0 ...

<Variance> 39

1.0 1.0 1.0 ...

<State> 3

<Mean> 39

0.0 0.0 0.0 ...

<Variance> 39

1.0 1.0 1.0 ...

<State> 4

<Mean> 39

0.0 0.0 0.0 ...

<Variance> 39

1.0 1.0 1.0 ...

<State> 5

<Mean> 39

0.0 0.0 0.0 ...

<Variance> 39

1.0 1.0 1.0 ...

<State> 6

<Mean> 39

0.0 0.0 0.0 ...

<Variance> 39

1.0 1.0 1.0 ...

<TransP> 5

0.0 1.0 0.0 0.0 0.0

0.0 0.6 0.4 0.0 0.0

0.0 0.0 0.6 0.4 0.0

0.0 0.0 0.0 0.7 0.3

0.0 0.0 0.0 0.0 0.0

<EndHMM>

6) Evaluating the recognizer against the test data
<p>[Whole]</p> <pre> ===== HTK Results Analysis ===== Date: Wed Apr 05 18:06:48 2006 Ref : testref.mlf Rec : recout.mlf ----- Overall Results ----- SENT: %Correct=76.72 [H=982, S=298, N=1280] WORD: %Corr=90.54, Acc=89.91 [H=4056, D=62, S=362, I=28, N=4480] </pre>
<p>[Phone]</p> <pre> ===== HTK Results Analysis ===== Date: Fri Apr 07 05:56:25 2006 Ref : testref.mlf Rec : recout.mlf ----- Overall Results ----- SENT: %Correct=95.23 [H=1219, S=61, N=1280] WORD: %Corr=98.62, Acc=98.62 [H=4418, D=0, S=62, I=0, N=4480] ===== </pre>

4.3 Result

The main probably reason behind the performance gap is due to the way HMM traditionally handles its acoustic modeling process, indiscriminate classification. This type of learning scheme suffers the most when the training data is insufficient to model each own classes precisely, hence the classifier often ‘confuses’ to generalize over new set of instances (the testing data).

We devised a simple approach to augment the size of training samples used by SVM. The motivation came from the nature of generic classification itself – no data like more data. It was also conceived due to the rigidity of using typical spectral analysis

with fixed time duration. The idea was applied in two areas: re-sampling and spectral analysis.

Apart from data reduction, the purpose for re-sampling the data into three smaller sub-samples is to provide the test sample with a virtual space in SVM's separating hyperplane. The same reason applies in spectral analysis counterpart for using three different time frames instead of just one. We re-sampled our own data from 16 kHz to: a) $\frac{1}{2}$ of 16 kHz (1st-tier), b) $\frac{1}{2}$ of $\frac{1}{2}$ of 16 kHz (2nd-tier), and c) $\frac{1}{2}$ of $\frac{1}{2}$ of $\frac{1}{2}$ of 16 kHz (3rd-tier). This will split the sample into three. For spectral analysis, we used MFCC with three sets of frame size: a) 180 samples window by 60 step-sizes (1st-tier), b) 240 samples window by 80 step-sizes (2nd-tier), and c) 300 samples window by 100 step-sizes (3rd-tier). Consecutively, each of the re-sampled speech is tripled once more.

Through trial and error, recognition accuracy on the Ti46 dataset only benefited from spectral analysis augmentation alone, unlike both augmentations (re-sampling and spectral analysis) for our own data. A straightforward explanation for this is Ti46's unusual 12.5 kHz sampling rate (w.r.t. ours' 16 kHz), which does not fit the re-sampling methodology of down-sampling the data in halves.

We ran two sets of experiment for each dataset: a) training: one, using 2nd-tier configuration while the other, using all-tiers setup (in re-sampling and spectral analysis), b) testing: only a single-tier parameter (in re-sampling and spectral analysis) for both datasets).

We substituted the standard gauge for performance measure (accuracy) with probability estimates accuracy. It is not the case of hard-margin classification; this adjusted score depicts the confidence of SVM's separating margin (from test example's origin to the decision boundary) in hyperspace.

On top of numbers and graphs, we managed to demonstrate this phenomenon via data visualization using GGobi's 2D-tour function. The only pre-requisite is to determine

the corresponding support vectors before plotting, which can be obtained from the SVM model itself.

Table 4.2: Confusion matrix and result analysis for
a) baseline and b) proposed techniques.

a)

	b	c	d	e	g	p	t	v	z	Correct	Error
b	211	0	21	10	0	0	0	14	0	211	45
c	0	249	0	0	0	1	1	4	1	249	7
d	29	1	207	5	0	2	5	7	0	207	49
e	6	0	5	241	0	0	1	3	0	241	15
g	0	1	6	0	246	0	3	0	0	246	10
p	3	1	1	0	2	236	8	5	0	236	20
t	0	1	0	0	3	1	250	1	0	250	6
v	20	1	22	3	1	6	5	195	3	195	61
z	1	5	0	0	0	0	2	7	241	241	15

Class	Sample	Correct	Error	Accuracy	Prob	ProbError	ProbAccy
b	256	211	45	82.42%	157.48	98.52	61.52%
c	256	249	7	97.27%	230.76	25.24	90.14%
d	256	207	49	80.86%	158.34	97.66	61.85%
e	256	241	15	94.14%	219.90	36.10	85.90%
g	256	246	10	96.09%	225.34	30.66	88.02%
p	256	236	20	92.19%	209.99	46.01	82.03%
t	256	250	6	97.66%	222.79	33.21	87.03%
v	256	195	61	76.17%	163.29	92.71	63.79%
z	256	241	15	94.14%	222.99	33.01	87.10%
Total	2304	2076	228	90.10%	1810.88	493.12	78.60%

Probability estimation score = **90.06%**

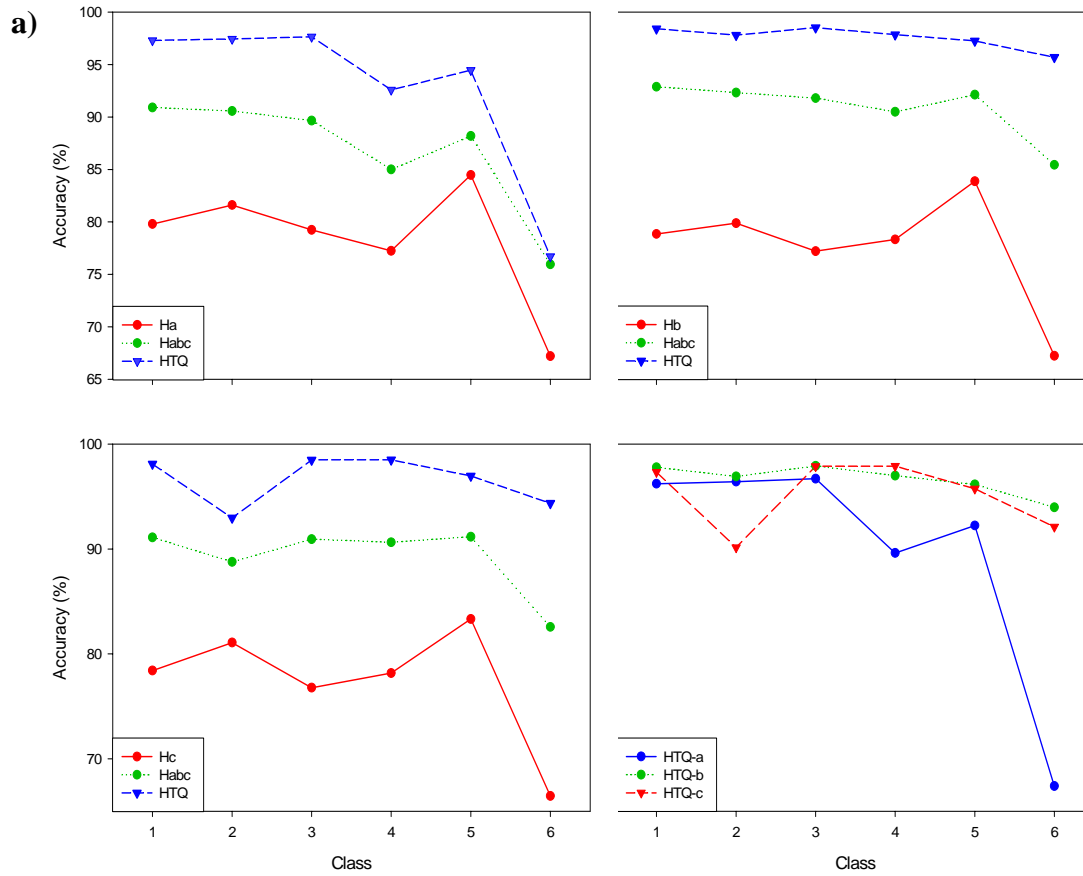
b)

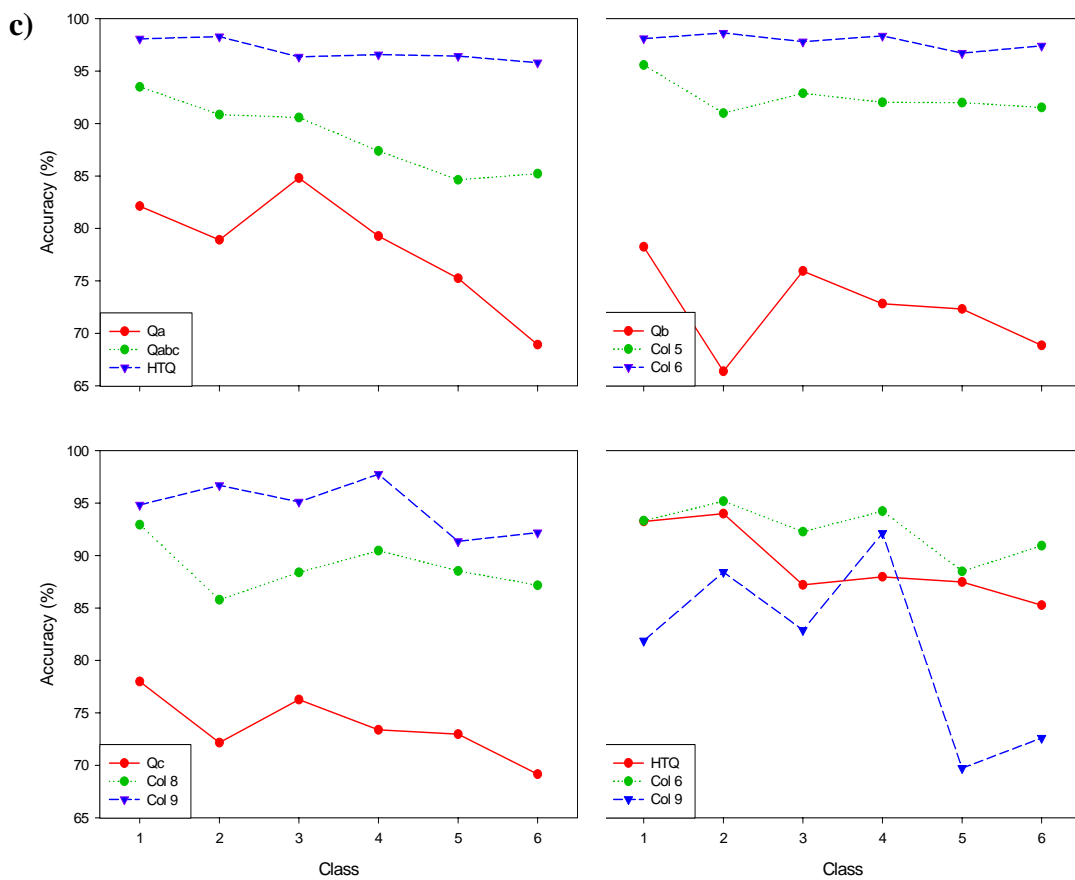
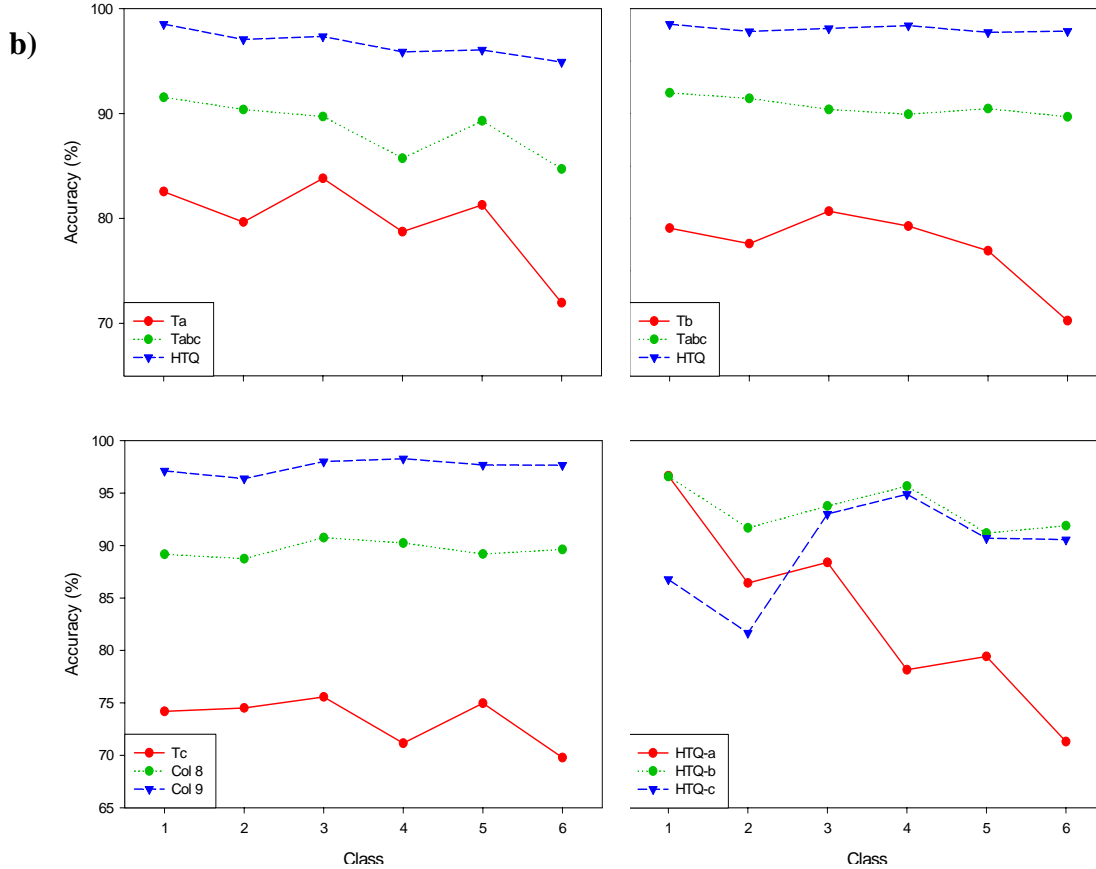
	b	c	d	e	g	p	t	v	z	Correct	Error
b	210	0	16	10	0	0	0	20	0	210	46
c	0	251	0	0	0	0	1	2	2	251	5
d	25	0	209	4	1	2	4	11	0	209	47
e	5	0	5	242	0	0	2	1	1	242	14
g	0	0	2	0	248	0	5	1	0	248	8
p	4	0	1	0	3	240	4	4	0	240	16
t	0	0	1	0	1	0	253	1	0	253	3
v	17	1	14	3	0	3	6	209	3	209	47
z	0	3	0	0	0	0	1	6	246	246	10

Class	Sample	Correct	Error	Accuracy	Prob	ProbError	ProbAccy
b	256	210	46	82.03%	176.33	79.67	68.88%
c	256	251	5	98.05%	239.93	16.07	93.72%
d	256	209	47	81.64%	179.62	76.38	70.16%
e	256	242	14	94.53%	230.06	25.94	89.87%
g	256	248	8	96.88%	237.08	18.92	92.61%
p	256	240	16	93.75%	226.96	29.04	88.66%
t	256	253	3	98.83%	241.53	14.47	94.35%
v	256	209	47	81.64%	187.93	68.07	73.41%
z	256	246	10	96.09%	239.36	16.64	93.50%
Total	2304	2108	196	91.49%	1958.79	345.21	85.02%

Probability estimation score = **91.47%**

N.B.: $Prob$ = *Correct's* probability estimation, $ProbError$ = *Error's* probability estimation, $ProbAccey$ = $Prob$ over *Sample*. Probability estimation score is obtained via proper class distribution weighting.





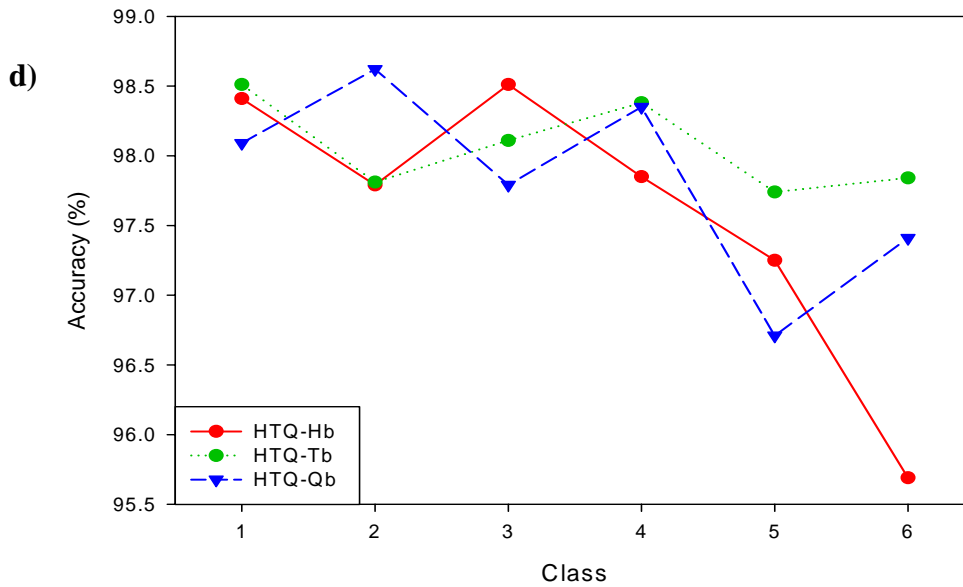
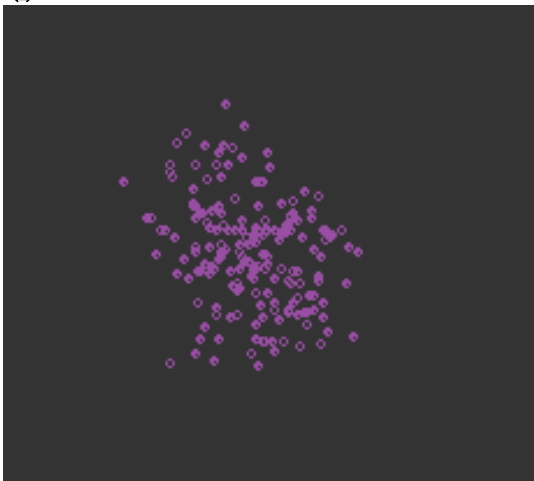


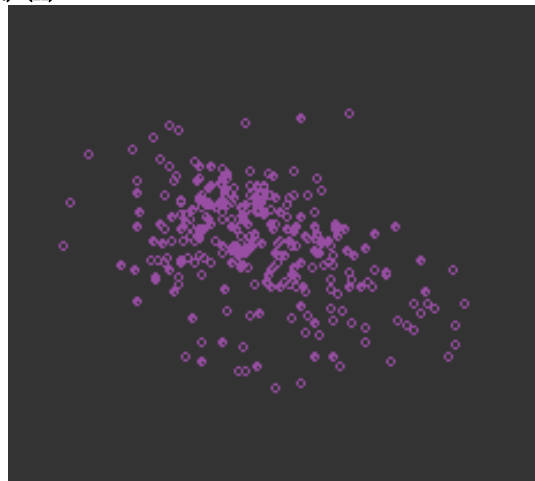
Figure 4.1: Performance comparison of the data augmentation technique.

N.B.: The graphs are generated from Table 4.2's *ProbAccy* column but they are based on the results of our own dataset. Re-sampling codes: H = 1st-tier, T = 2nd-tier, Q = 3rd-tier; Spectral analysis codes: a = 1st-tier, b = 2nd-tier, c = 3rd-tier. Experiment setups: a) (i) Ha on Ha, Habc, HTQ; (ii) Hb on Hb, Habc, HTQ; (iii) Hc on Hc, Habc, HTQ; (iv) the best of (i), (ii), (iii); with HTQ = all-tiers setup. Same settings are applied to b) and c) w.r.t. T and Q. In conclusion, from d) the best of (a), (b), (c), testing Tb on HTQ train configuration outperforms the rest.

a) (i)



a) (ii)



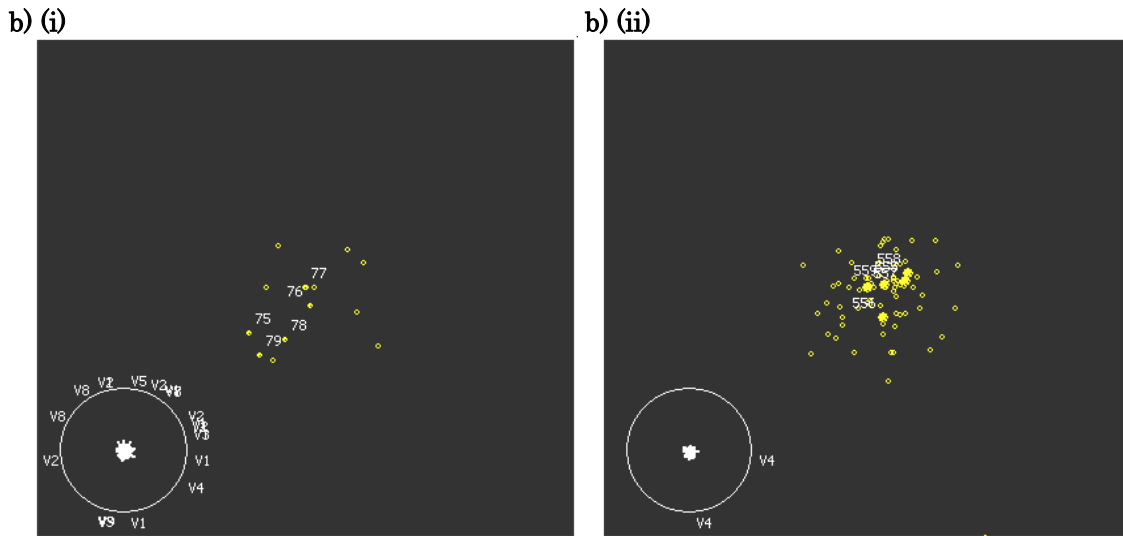


Figure 4.2: Data visualization using GGobi for **a)** letter *b* and **b)** word *sorot*

N.B.: The plots depicted the following: support vectors (hallow circles) of the train data, and test examples (dots), while the rest of the train data are omitted. Left side represents partial result of Table 4.2 (a) while the other denotes Table 4.2 (b). It can be seen that the test examples in the right view are more centralized towards the core of the support vectors, resulting in a much larger margin separation. This explains why *ProbAccy* of Table 4.2(b) is higher than Table 1(a); it proves that the separating hyperplane is more confidence.

Table 4.3: Recognition performance benchmark

Method		Recognition Accuracy			
		<i>D</i>	<i>S</i>	<i>I</i>	%
HTK's HMM	<i>Word</i>	0	0	0	100 (280/280)
	<i>Sent</i>	-	-	-	100 (80/80)
SVM-DSW	<i>Word</i>	0	1	0	99.64 (279/280)
	<i>Sent</i>	-	1	-	98.75 (79/80)

4.4 Demo

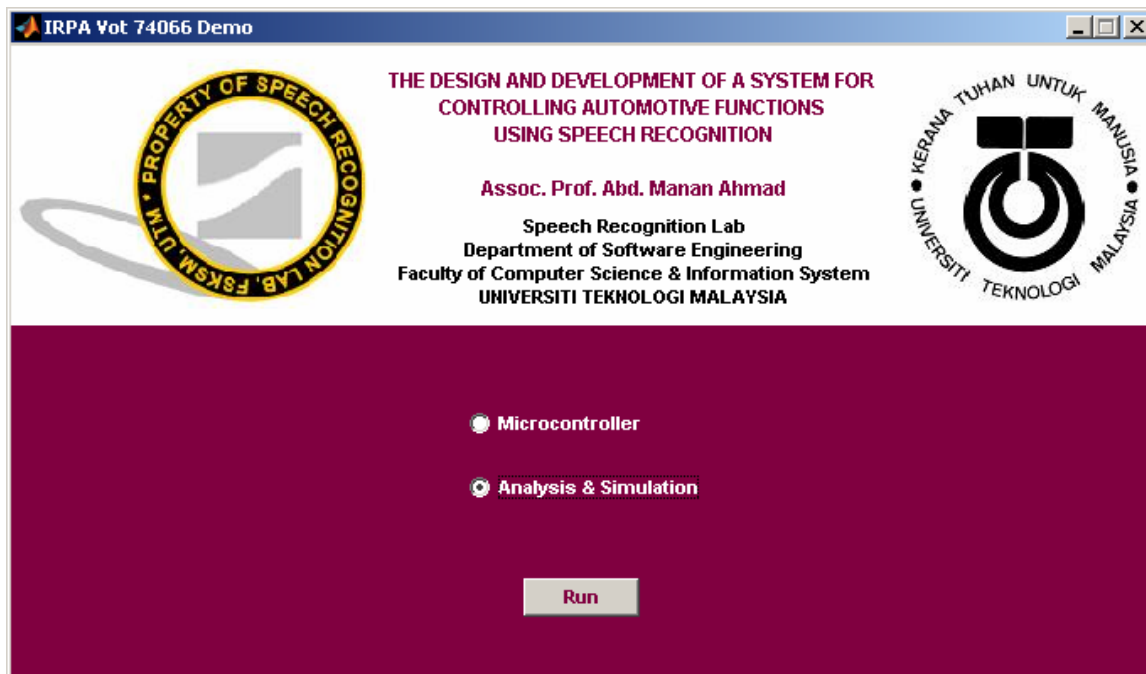


Figure 4.3: Controlling Automotive Functions using Speech Recognition GUI

For a simple demo to apply voice automotive controlling device, there are an interface for a user to input the command using their voice. After some data processing, the result will be display on screen. The action of the automotive controlling function will be displays as a graphic demonstration.

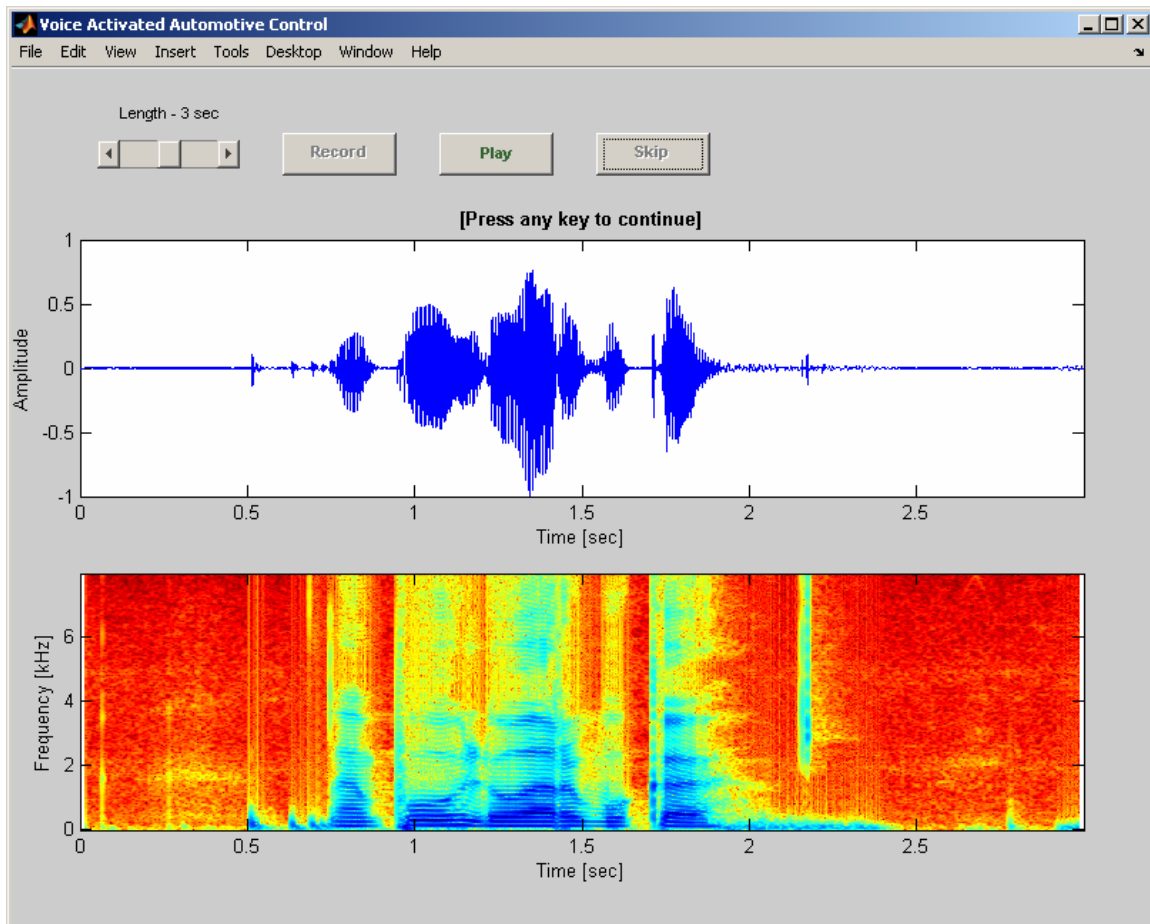


Figure 4.4: Recording voice using Matlab interface for “lampu luar buka”.

Figure 4.4 show the interfaces that get the input from the user. The input was human voice to control automotive devices like radio, air-cond, wiper and etc. For an example, the input for this demo was ‘*lampu luar buka*’. With that command, that spoke in Malay will make the consumer’s car switch on the head lamp.

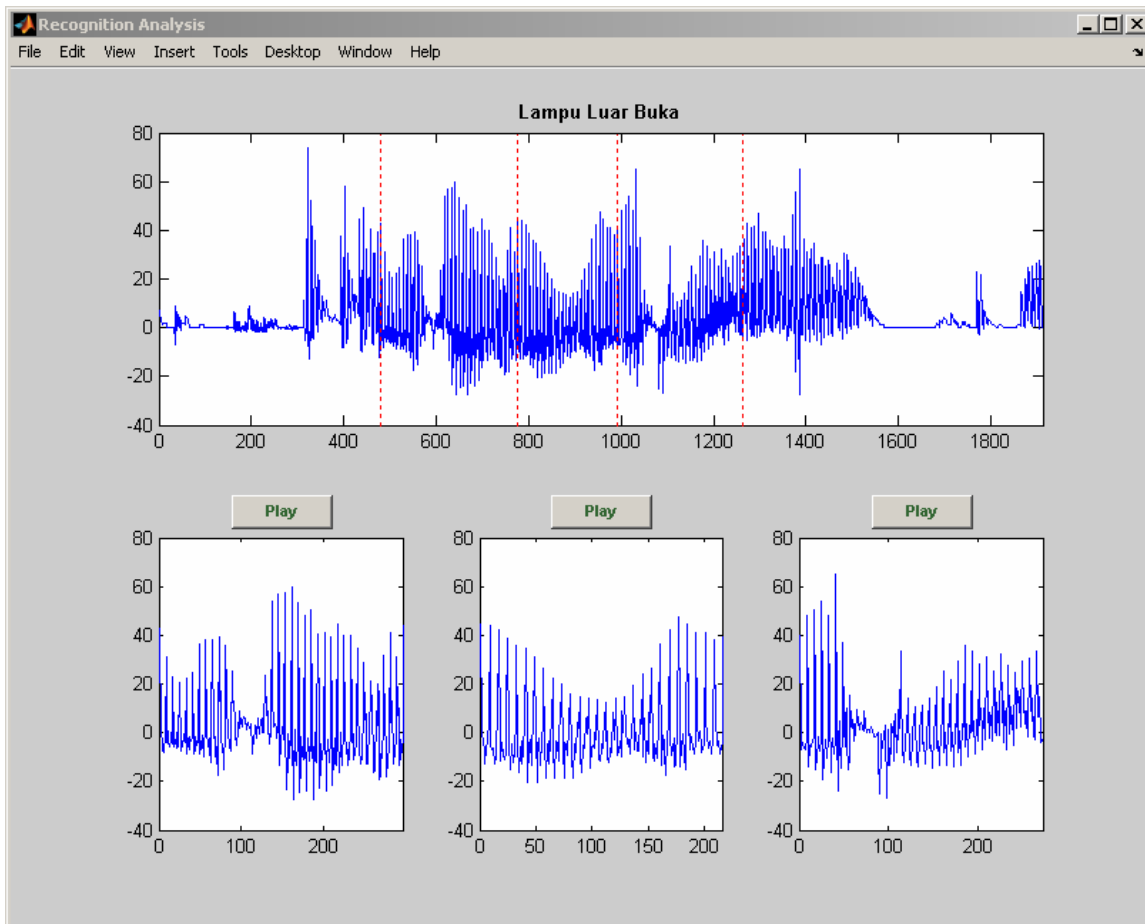


Figure 4.5: Wav file for “lampu luar buka”

After the human voice has been recorded and save in wav file format, the file will be segmented. As been show above, the “lampu luar buka” has been segmented into three continuous words.

To view the currently device status, there is an interface that responsible to do that purposed. Figure 4.6 shows the currently status of every automotive controlling devices for a car that involved in this research.

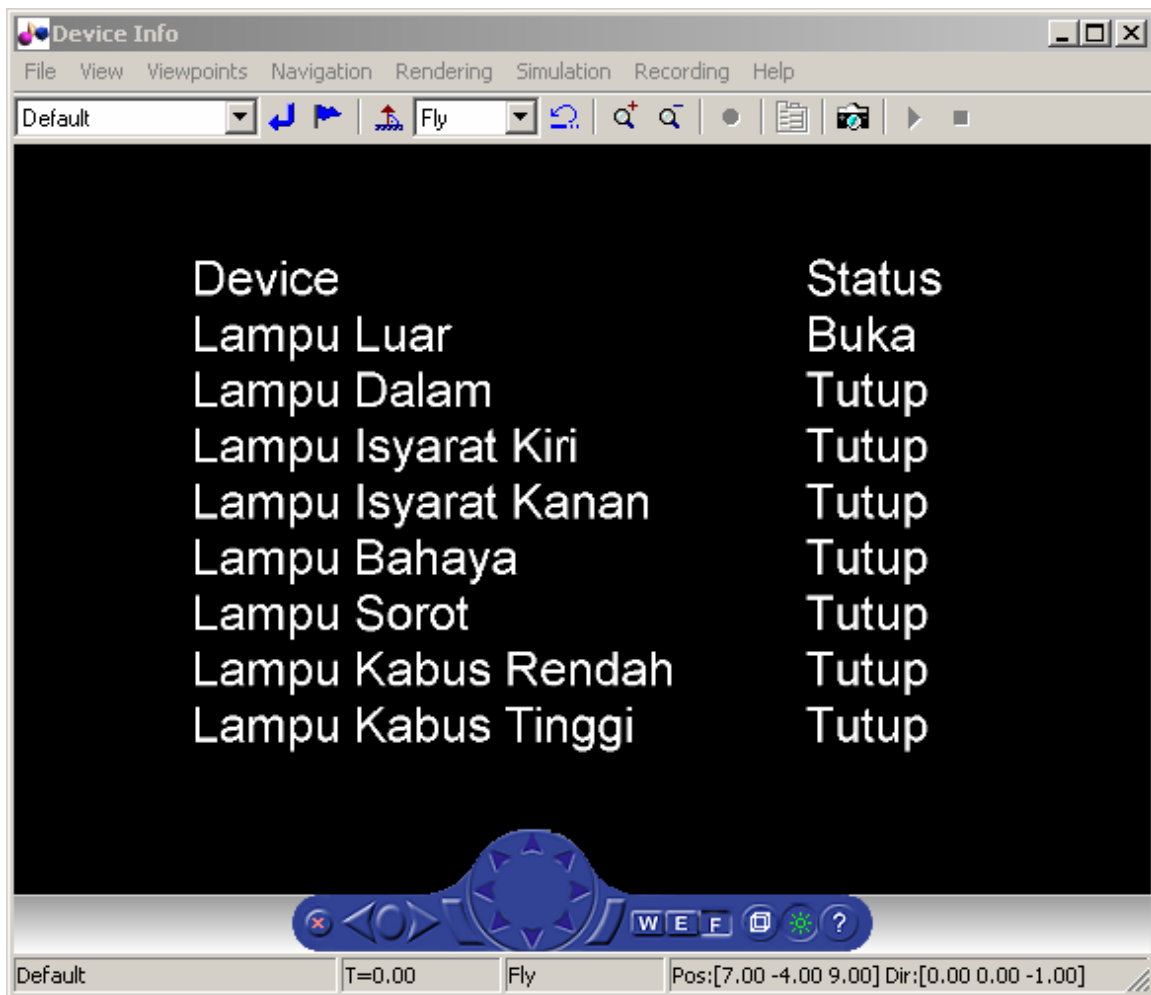


Figure 4.6: Device info for “lampu luar buka”

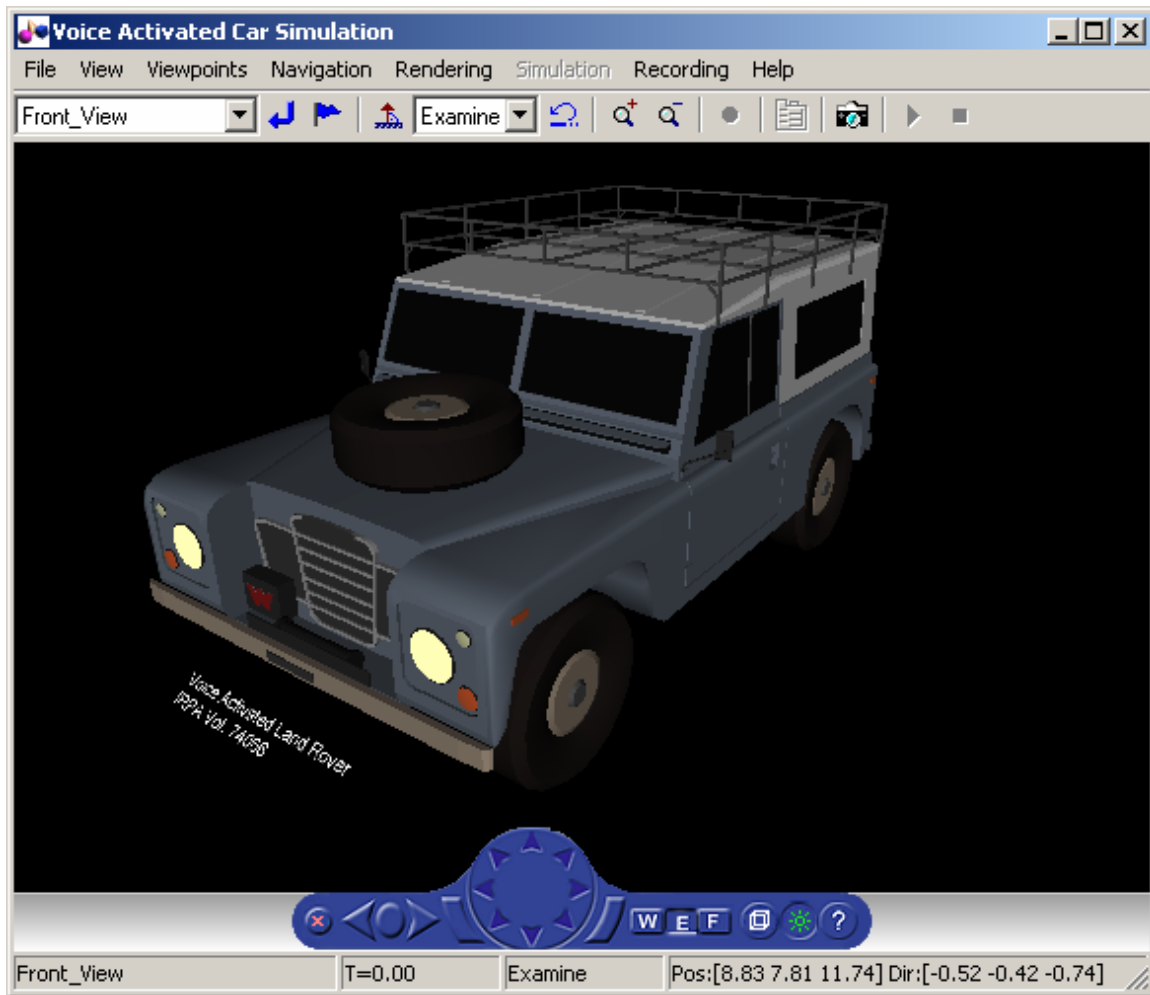


Figure 4.7: Simulation for “lampu luar buka”

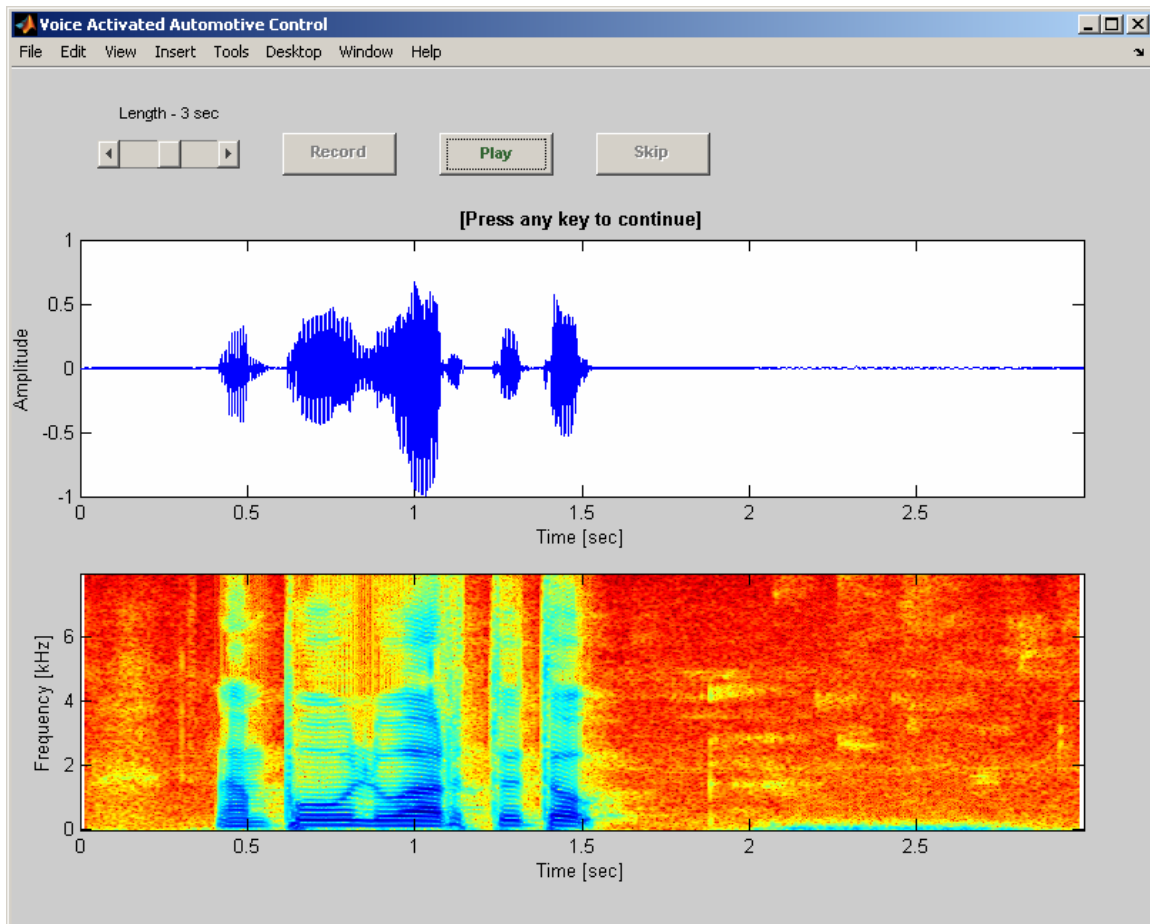


Figure 4.8: Recording voice using Matlab interface for “lampu luar tutup”.

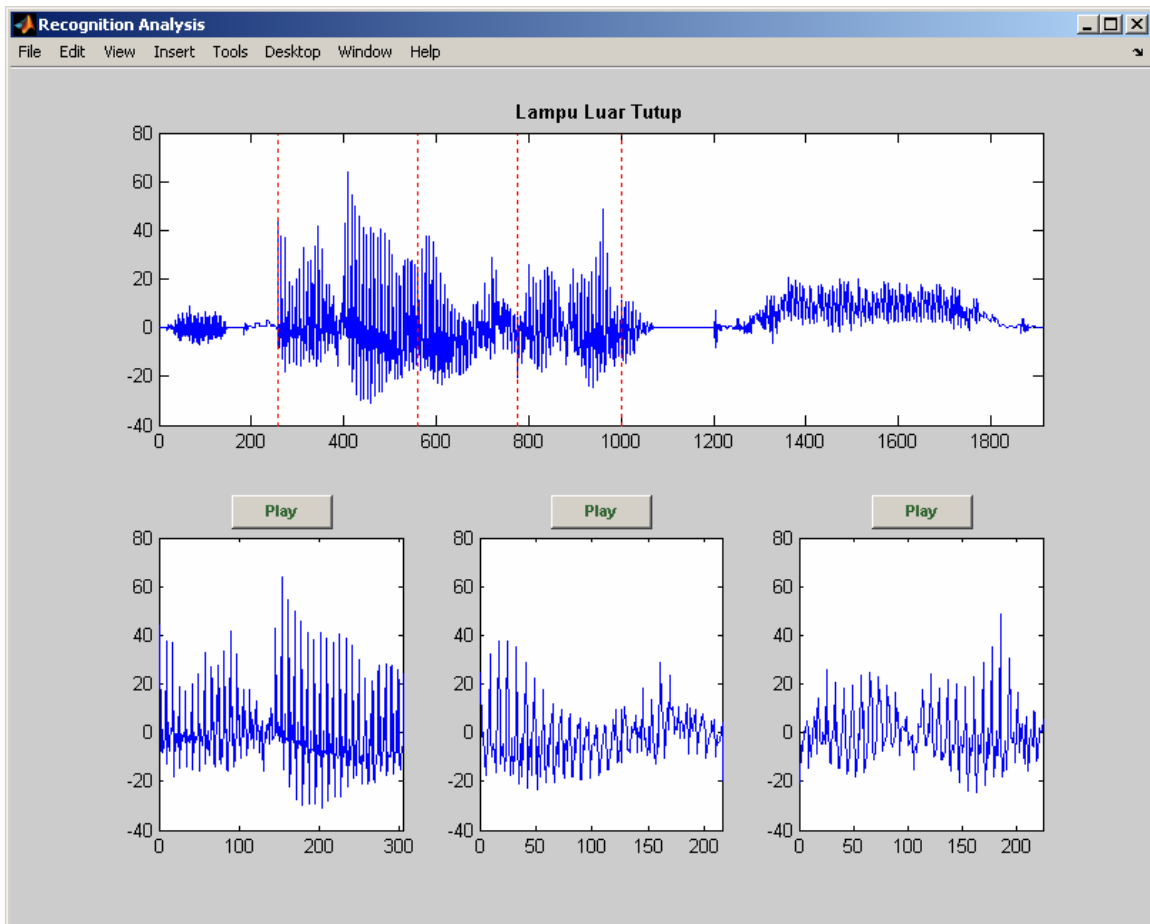


Figure 4.9: Wav file for “lampu luar tutup”

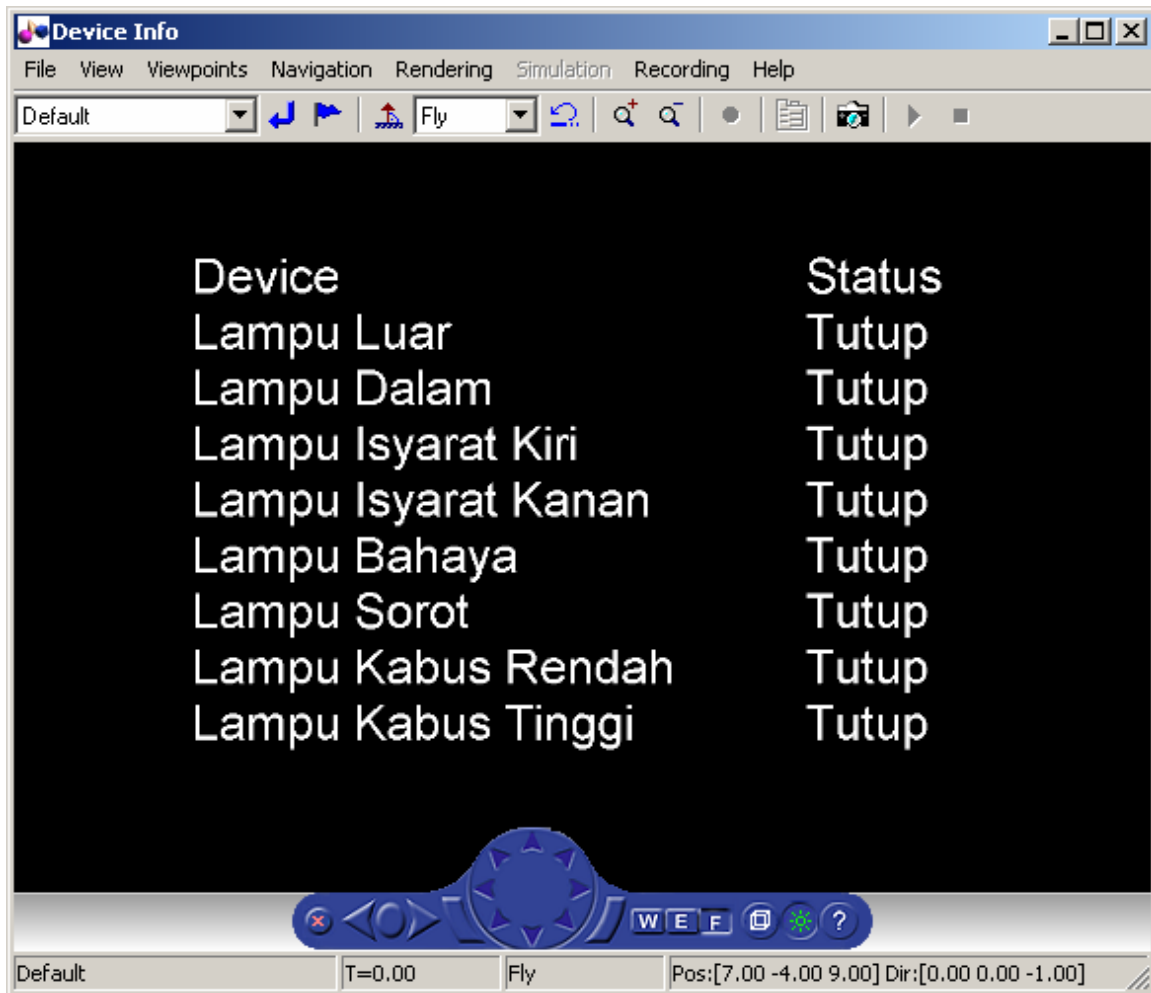


Figure 4.10: Device info for “lampu luar buka”

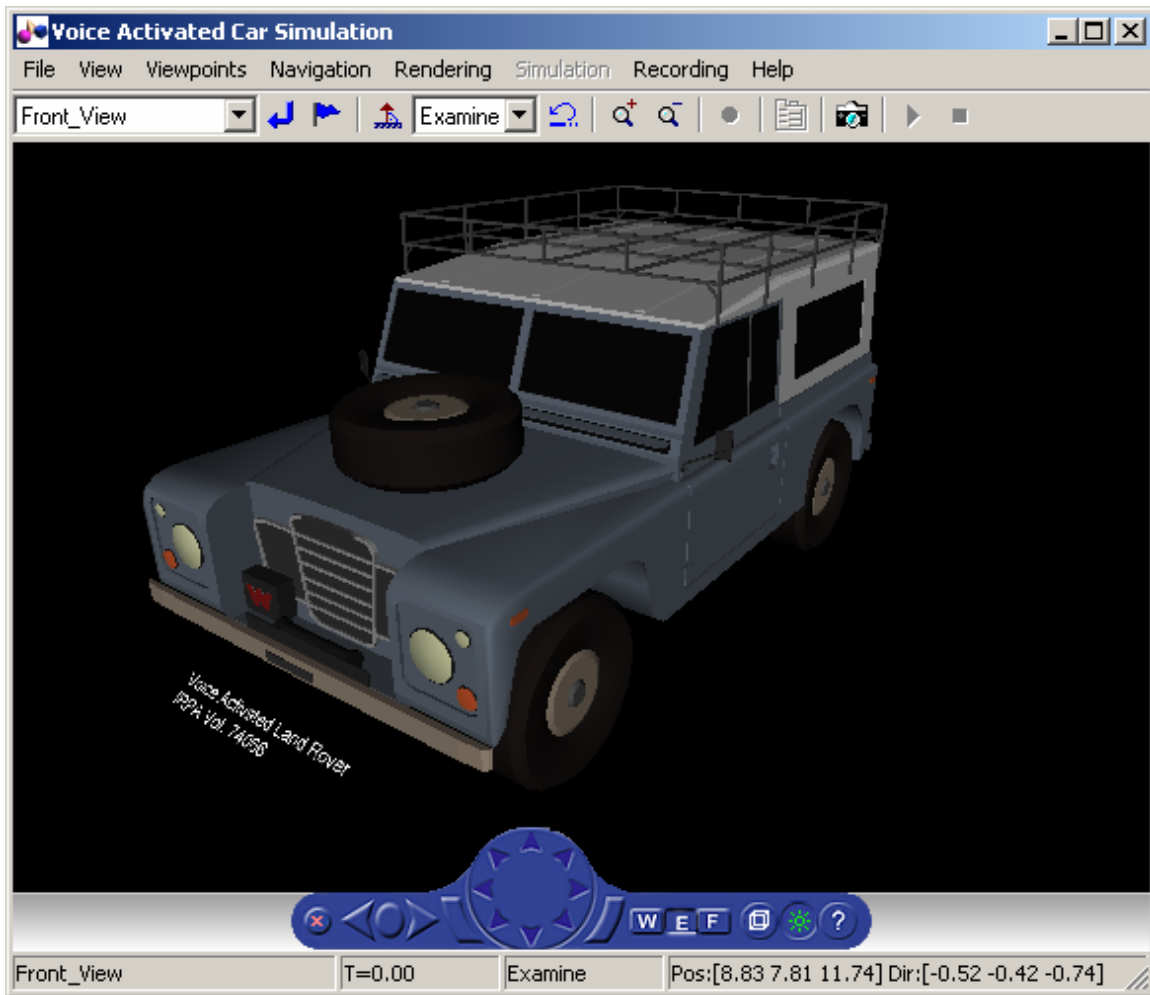


Figure 4.11: Simulation after “lampu luar tutup” recognition process

4.6 Summary

Data for this research has been described detail in this chapter. It described how the data managed appropriately to gain a good data for research purposed. In this chapter also, the result from the experiments that been done has been described. It show the performance of the DSR-DSW speech recognition engine to recognize human voice and from the recognition result had been took as an input for automotive controlling devices commanding.

CHAPTER V

CONCLUSION AND SUGGESTION

5.1 Introduction

A design and development of a system for controlling automotive function using speech recognition already has been developed. A system that received a voice in Malay's spoken as an input to control such devices in a car (radio, air-conditioner, wiper and etc). There are 9 devices that can be control and 69 commands for those devices. The speech recognition engine namely SVM-DSW is responsible for speech processing while the output can be shows as visualization using GGobi tools.

The project began with understanding the basic requirements for developing a speech recognition-based application. The system consists of a speech recognizer engine and a speech database in Malay, which corresponds to a complete set of automotive controls. Both components were built concurrently. In summary, there are 69 commands (sentences) for controlling 9 automotive devices with a total of 48 words involving 8 speakers from each gender. Speech acquisition task was done under considerable controlled condition (approximately 30dB of signal-to-noise ratio) with noise-cancellation headset microphones. Each subject was instructed to speak continuously into

the microphone with 10 repetitions per utterance. We then allocated the whole dataset evenly for experimentation; 5 repetitions for training and testing respectively. Manual segmentation (word annotation of the speech sample) was applied only for the training set in order to generate corresponding acoustic models.

The recognition engine incorporates the Support Vector Machines (modified from LibSVM, a library for SVM) classifier scheme with Dynamic Shifting Window (DSW) as a complement for SVM's inability to handle time-varying data such in speech utterance. Justification for using SVM is largely due to its powerful discriminative trait which surpasses that of Artificial Neural Network's (ANN). Inspiringly, through experiments, SVM-DSW even outperforms the predominant technique in speech recognition, namely the Hidden Markov Models (applying Cambridge University's Hidden Markov Toolkit - HTK). The gist of our work is the development of DSW method which enables the SVM to be robust against silence and noise factors that may exist along the path of speech signal as well as handling the variability of speaking rate.

In the final leg of project, we simulate the whole process of controlling automotive functions via voice. Matlab is used as the main platform for real-time testing (recognition) and output analysis (segmentation). The car simulation is produced by Matlab's Virtual Reality Toolbox.

5.2 Advantages

Several advantages can be gained from SVM-DSW for controlling automotive devices using speech recognition development research. The advantages that can be gained from the research such as:

- i. Developing indigenous technology for local based sectors in the automotive industry.

- ii. Provide leverage to the local automotive industry by introducing competitive technology to the sector.
- iii. Encouraging safer driving methods through promoting hands-free tuning of controls.
- iv. Domestic industry linkages – this project will establish a collaboration between the researchers of UTM and Proton as well as DBP.

5.2.1 Commercialization

As car manufacturer installed more electronic control interface like wifi, bluetooth or infrared using CAN Bus system, the control of certain device in the car can be transfer form the car to another device such as Personal Digital Assistant (PDA) provide the cars manufactures giving the support to the development. As PDA is a complete computer but much smaller scale, it can be mount on the car's dashboard. The PDA is then installed with speech recognition system, making the car devices can be control by speech. For best performance and security issues the user need to use a Bluetooth headset as interface to the PDA. Speech engine in the PDA is giving many advantages such as the easy of upgradeability of the speech engine, easier testing and verifying and mobility as the user may has two cars, the system easily can be installed to the other car without much hassle.

The SVM-DSW recognizer engine is targeted for applications that demand accuracy and reliability. Accuracy gets top priority in high risk tasks such as vehicle manoeuvring, surgical procedures, etc. where the slightest error gives disastrous consequences. Consistency is also of paramount prerequisite because the accuracy has to be reproducible time and time again without failure. SVM-DSW has both of these qualities as well as being low in computational cost (using whole word recognition unit and embedded grammar rule) allows it to be ported into Very Large Scale Integration (VLSI) technology. Voice activated household appliances could also benefit from such integration.

5.2.2 Potential Beneficiaries

The project produces a device for car control and speech database (bahasa Melayu) first in the country. The speech database can be used as a standard so that any speech engine develop for Malay can be tested for accuracy and comparison can be make with other speech engine. So, two areas that can involve using this SVM-DSW engine are:

- i. Perushaan Otomobil Nasional (PROTON) – Automative Control System
- ii. Dewan Bahasa dan Pustaka – Automative Control Carpus (Speech Database)

However, these two areas just potential beneficiaries that related with this engine. The engine also can be applied to another area and purpose based on the domain.

5.3 Summary

This research project ‘The Design and Development of a System for Controlling Automotive Function using Speech Recognition’ began with understanding the basic requirements for developing a speech recognition-based application. The system consists of a speech recognizer engine and a speech database in Malay, which corresponds to a complete set of automotive controls. Both components were built concurrently.

The recognition engine incorporates the Support Vector Machines (modified from LibSVM, a library for SVM) classifier scheme with Dynamic Shifting Window (DSW) as a complement for SVM’s inability to handle time-varying data such in speech utterance. Justification for using SVM is largely due to its powerful discriminative trait which surpasses that of Artificial Neural Network’s (ANN). Inspiringly, through experiments, SVM-DSW even outperforms the predominant technique in speech recognition, namely the Hidden Markov Models (applying Cambridge University’s

Hidden Markov Toolkit - HTK). The gist of our work is the development of DSW method which enables the SVM to be robust against silence and noise factors that may exist along the path of speech signal as well as handling the variability of speaking rate.

For a result of this research, we simulate the whole process of controlling automotive functions via voice. Matlab is used as the main platform for real-time testing (recognition) and output analysis (segmentation). The car simulation is produced by Matlab's Virtual Reality Toolbox. The result from the experiments shows excellent contribution to the speech recognition engine performance.

REFERENCES

- Abd Manan Ahmad, Ag. Noorajis Ag. Nordin, Emrul Hamide Md. Saa'im, Den Fairol bin Samaon and Mohd Danial bin Ibrahim, *An Architecture Design of the Intelligent Agent for Speech Recognition and Translation*, 14th International Conference on Computer Theory and Applications (ICCTA' 2004), Sheraton Hotel, Alexandria, Egypt. 28 – 30 September 2004.
- BENGIO, Y. 1993. A Connectionist Approach to Speech Recognition. *International Journal on Pattern Recognition and Artificial Intelligence*, 7, 647-668.
- BURGES, C. 1998. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.
- CHANG, C.C., AND LIN, C.J. 2001. LIBSVM: A Library for Support Vector Machines (Version 2.6). <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CHIN, K.K. 1999. *Support Vector Machines Applied to Speech Pattern Classification*. Master thesis, Cambridge University.
- COOK, G.D. 1997. *Data Selection and Model Combination in Connectionist Speech Recognition*. PhD thesis, Cambridge University.

- CORTES, C., AND VAPNIK, V. 1995. Support-Vector Networks. *Machine Learning*, 20, 273-297.
- COSI, P., HOSOM, J.P., SCHALKWYK, J., SUTTON, S., COLE, R.A. 1998. Connected Digit Recognition Experiments with the OGI Toolkit's Neural Network and HMM-Based Recognizers. In *Proceedings of 4th IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, Turin Italy*, 135-140.
- DELIGNE, S., DHARANIPRAGADA, S., GOPINATH, R., MAISON, B., OLSEN, P., AND PRINTZ, H. 2002. A Robust High Accuracy Speech Recognition System for Mobile Applications. In *Proceedings of IEEE Trans. on Speech and Audio Processing*, 10, 8, 551-561.
- Den Fairol Samaon, Abdul Manan Ahmad, Md Sah Salam, *Continuos Voice Activated Automotive Control via Sequential Feature Segmentation*, 2nd National Conference on Computer Graphics and Multimedia, ESSET, Selangor, Malaysia. 8 -10 December 2004.
- DUAN, K., KEERTHI, S.S., AND POO, A.N. 2003. Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters. *Neurocomputing*, 51, 41-59.
- ECK, D., GRAVES, A., AND SCHMIDHUBER, J. 2003. A New Approach to Continuous Speech Recognition Using LSTM Recurrent Neural Networks. *Technical Report IDSIA-14-03*, IDSIA. <http://www.idsia.ch/techrep.html>.
- Emrul Hamide Md Saaim, Ag Noorajis Ag. Nordin, Abdul Manan Ahmad, *Design of the Intelligent Agent for Speech Recognition Based on Web Application*, 2nd National Conference on Computer Graphics and Multimedia, ESSET, Selangor, Malaysia. 8 -10 December 2004.

Emrul Hamide Md Saaim, Abd Manan Ahmad, Mohamad Ashari Alias, *Applying Mobile Agent Technology in Distributed Speech Recognition*, Postgraduate Annual Research Seminar PARS'05 17-18 May, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.

Emrul Hamide Md Saaim, Abd Manan Ahmad, Mohamad Ashari Alias, Mohd Fauzi Othman, *Applying Mobile Agent in Distributed Speech Recognition using JADE*, 1st International Conference on Computer, Communication and Signal Processing with Special Track on Biomedical Engineering, Renaissance Hotel Kuala Lumpur, Malaysia, 14-16 November 2005.

FURUI, S. 2003. Toward Spontaneous Speech Recognition and Understanding. *Pattern Recognition in Speech and Language Processing*, W. Chou and B.H. Juang, Eds., CRC Press LLC, New York, 191-227.

GANAPATHIRAJU, A. 2002. *Support Vector Machines for Speech Recognition*. PhD thesis, Mississippi State University.

GOH KIA ENG, ABD MANAN AHMAD, *A 3-State Endpoint Detection Algorithm using Multiple Features*, The 6th IASTED International Conference on Signal and Image Processing ~SIP2004~ August 23 – 25, 2004 Honolulu, Hawaii, USA

GOH KIA ENG, ABD MANAN AHMAD, *A 3-Level Endpoint Detection Algorithm for Isolated Speech using Time and Frequency-Based Features*, 2004 International Conference on Control, Automation and Systems (ICCAS 2004), August 25 – 27, 2004, The Shangri-La Hotel, Bangkok, Thailand.

Goh Kia Eng and Abd Manan Ahmad, *Malay Syllable Speech Recognition using Hybrid Neural Network*, International Conference on Control, Automation, and Systems (ICCAS2005), June 2-5, 2005, Korea International Exhibition Center) The Province of Gyeonggi, Korea.

- Goh Kia Eng and Abd Manan Ahmad, *Malay Speech Recognition using Self-Organizing Map and Multi-layer Perceptron*, Postgraduate Annual Research Seminar PARS'05 17-18 May, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.
- HAFFNER, P., FRANZINI, M., AND WAIBEL, A. 1991. Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. In *Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 105-108.
- HILD, H. AND WAIBEL, A. 1993. Connected Letter Recognition with a Multi-State Time Delay Neural Network. In *Neural Information Processing Systems 5*, S. Hanson, J. Cowan and C.L. Giles, Eds., Morgan Kaufmann.
- HOSOM, J.P., COLE, R.A, AND COSI, P. 1998. Improvements in Neural-Network Training and Search Techniques for Continuous Digit Recognition. *Australian Journal of Intelligent Information Processing Systems*, 5, 4, 277-284.
- HSU, C.W., AND LIN, C.J. 2002. A Comparison of Methods for Multi-class Support Vector Machines. In *Proceedings of IEEE Trans. on Neural Networks*, 13, 415-425.
- HUERTA, J. M. 2000. Speech Recognition in Mobile Environments. PhD thesis, Carnegie Mellon University.
- JELINEK, F. 1976. Continuous Speech Recognition by Statistical Methods. In *Proceedings of IEEE*, 64, 4, 532-556.
- KIRSCHING, I., TOMABECHI, H., KOYAMA, M., AND AOE, J.I. 1996. The Time-Sliced Paradigm - A Connectionist Method for Continuous Speech Recognition. *Information Sciences*, 93, 1, 133-158.

- LIN, H.T., LIN, C.J., AND Weng, R.C. 2003. A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Technical Report*, Department of Computer Science and Information Engineering, National Taiwan University.
- M. Masroor Ahmed and Abd Manan Ahmad, *Review and Challenges in Speech Recognition*, International Conference on Control, Automation, and Systems (ICCAS2005), June 2-5, 2005, Korea International Exhibition Center) The Province of Gyeonggi, Korea.
- MA, C., RANDOLPH, M.A., AND DRISH, J. 2001. A Support Vector Machines-based Rejection Technique for Speech Recognition. In *Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 1, 381-384.
- MCDERMOTT, E. AND KATAGIRI, S. 1991. LVQ-based Shift-Tolerant Phoneme Recognition. In *Proceedings of IEEE Trans. on Acoustics, Speech, and Signal Processing*, 39, 6, 1398-1411.
- Mohd Danial Ibrahim, Abd Manan Ahmad, Den Fairol Samaon and Md Sah Salam, *Discriminative Approach for E-Set Recognition using Time-Expanded Features*, 14th International Conference on Computer Theory and Applications (ICCTA' 2004), Sheraton Hotel, Alexandria, Egypt. 28 – 30 September 2004.
- Mohd Danial Ibrahim, Abdul Manan Ahmad, Den Fairol Samaon, Md Sah Salam, *Improved E-Set Recognition Performance using Time-Expanded Features*. 2nd National Conference on Computer Graphics and Multimedia, ESSET, Selangor, Malaysia. 8 -10 December 2004.
- MOLAU, S., PITZ, M., SCHLUTER, R., AND NEY, H. 2001. Computing Mel-Frequency Cepstral Coefficients on the Power Spectrum. In *Proceedings of IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 1, 73-76.

- ROBINSON, A.J., COOK, G.D., ELLIS, D.P.W., FOSLER-LUSSIER, E., RENALS, S.J., WILLIAMS, D.A.G. 2002. Connectionist Speech Recognition of Broadcast News. *Speech Communication*, 37, 1-2, 27-45.
- SALAM, M.S. 2001. *Pengoptimuman Rangkaian Neural Menggunakan Algoritma Genetik Dalam Pengecaman Suara*. Master thesis, Universiti Teknologi Malaysia.
- SALIZA ISMAIL AND ABD MANAN AHMAD, *Reccurent Neural Network With Backpropagation Through Time Algorithm for Arabic Recognition*, 2004 International Conference on Control, Automation and Systems (ICCAS 2004), August 25 – 27, 2004, The Shangri-La Hotel, Bangkok, Thailand.
- SALIZA ISMAIL AND ABD MANAN AHMAD, *Reccurent Neural Network With Backpropagation Through Time Learning Algorithm for Arabic Phoneme Recognition*, ESM 2004, 18th European Simulation Multiconference June 13th – 16th, 2004, Magdeburg, Germany
- SALOMON, J. 2001. *Support Vector Machines for Phoneme Classification*. Master thesis, University of Edinburgh.
- SANCHEZ A, V.D. 2003. Advanced Support Vector Machines and Kernel Methods. *Neurocomputing*, 55, 1-2, 5-20.
- SMITH, N.D., AND GALES, M.J.F. 2002. Speech Recognition using SVMs. In *Neural Information Processing Systems*, 14, T.G. Dietterich, S. Becker, and Z. Ghahramani, Eds., MIT Press.
- TEBELSKIS, J. 1995. *Speech Recognition using Neural Networks*. PhD thesis, Carnegie Mellon University.

WET, F., CRANEN, B. VETH, J., AND BOVES, L. 2001. A Comparison of LPC and FFT-based Acoustic Features for Noise Robust ASR. In *Proceedings of Eurospeech 2001*, 865-868.

ZAVALIAGKOS, G., ZHAO, Y., SCHWARTZ, R., AND MAKHOUL, J. 1994. A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition. In *Proceedings of IEEE Trans. on Speech and Audio Processing*, 2, 1, 151-160.

APPENDIX

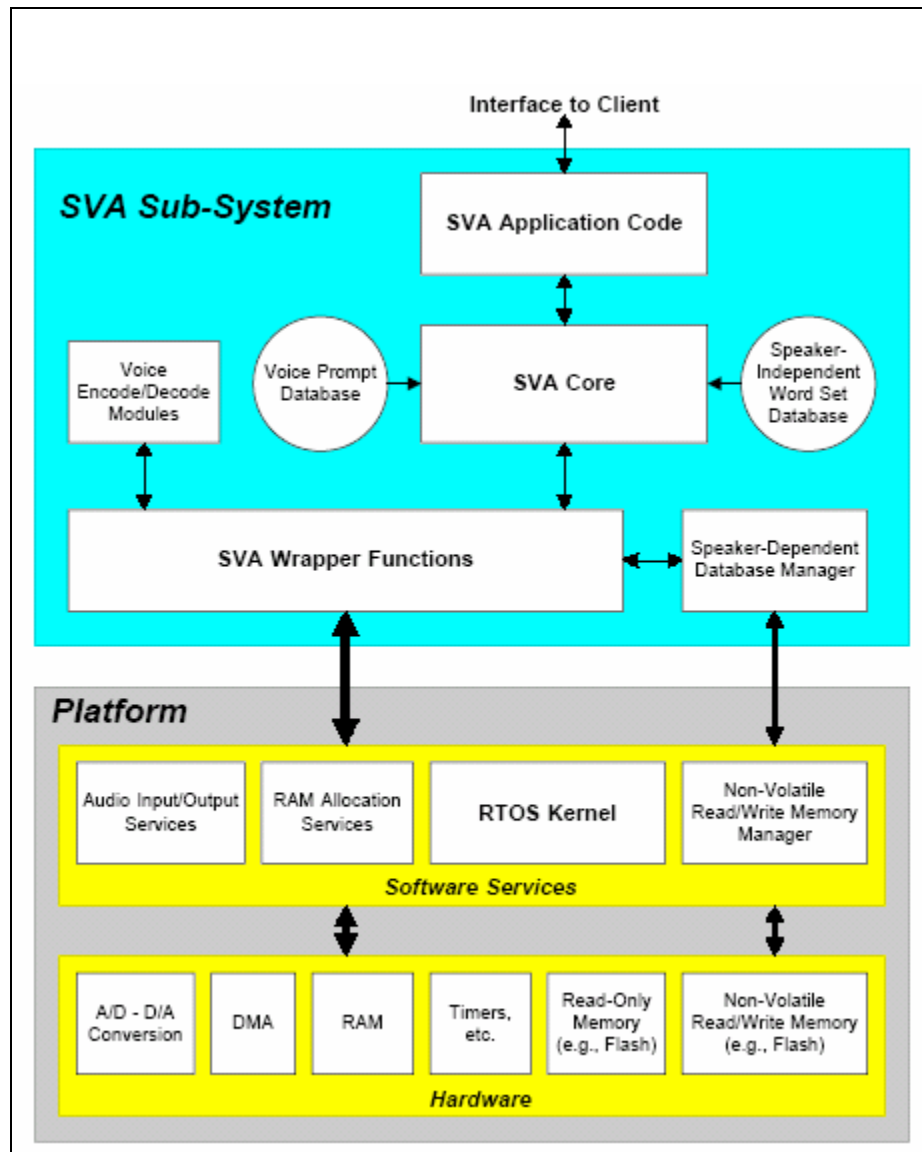
APPENDIX A : List of commands

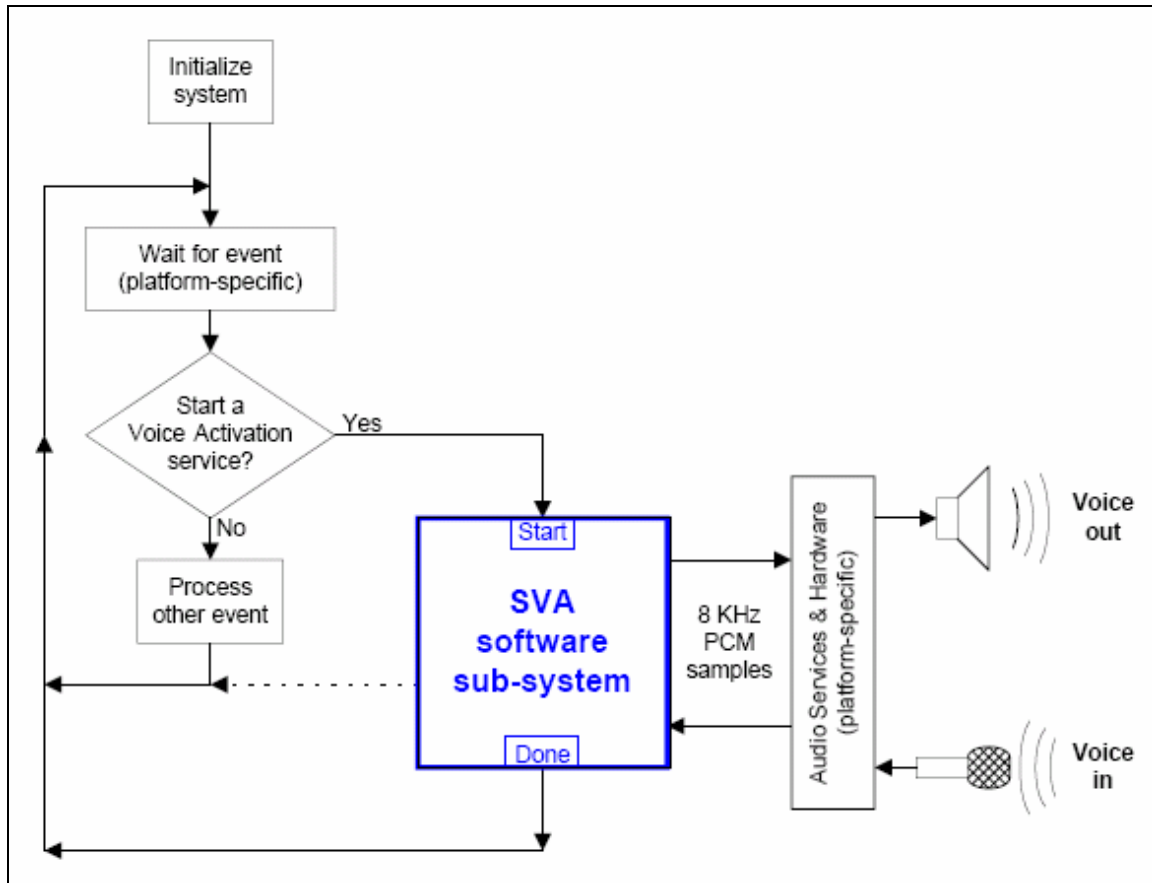
No.	List of Commands			
1	Lampu	Luar	Buka	
2	Lampu	Luar	Tutup	
3	Lampu	Dalam	Buka	
4	Lampu	Dalam	Tutup	
5	Lampu	Isyarat	Kiri	Buka
6	Lampu	Isyarat	Kiri	Tutup
7	Lampu	Isyarat	Kanan	Buka
8	Lampu	Isyarat	Kanan	Tutup
9	Lampu	Bahaya	Buka	
10	Lampu	Bahaya	Tutup	
11	Lampu	Sorot	Buka	
12	Lampu	Sorot	Tutup	
13	Lampu	Kabus	Rendah	Buka
14	Lampu	Kabus	Rendah	Tutup
15	Lampu	Kabus	Tinggi	Buka
16	Lampu	Kabus	Tinggi	Tutup
17	Gear	Satu		
18	Gear	Dua		
19	Gear	Pandu		
20	Gear	Neutral		
21	Gear	Undur		
22	Gear	Berhenti		
23	Pendingin	Buka		
24	Pendingin	Tutup		
25	Pendingin	Kipas		

26	Pendingin	Hawa	Satu	
27	Pendingin	Hawa	Dua	
28	Pendingin	Hawa	Tiga	
29	Tingkap	Depan	Kanan	Buka
30	Tingkap	Depan	Kanan	Tutup
31	Tingkap	Depan	Kiri	Buka
32	Tingkap	Depan	Kiri	Tutup
33	Tingkap	Belakang	Kanan	Buka
34	Tingkap	Belakang	Kanan	Tutup
35	Tingkap	Belakang	Kiri	Buka
36	Tingkap	Belakang	Kiri	Tutup
37	Pengilap	Buka		
38	Pengilap	Tutup		
39	Pengilap	Air		
40	Pengilap	Satu		
41	Pengilap	Dua		
42	Hon			
43	Radio	Buka		
44	Radio	Tutup		
45	Radio	Perlahan		
46	Radio	Kuat		
47	Radio	FM		
48	Radio	CD		
49	Radio	CD	Main	
50	Radio	CD	Berhenti	

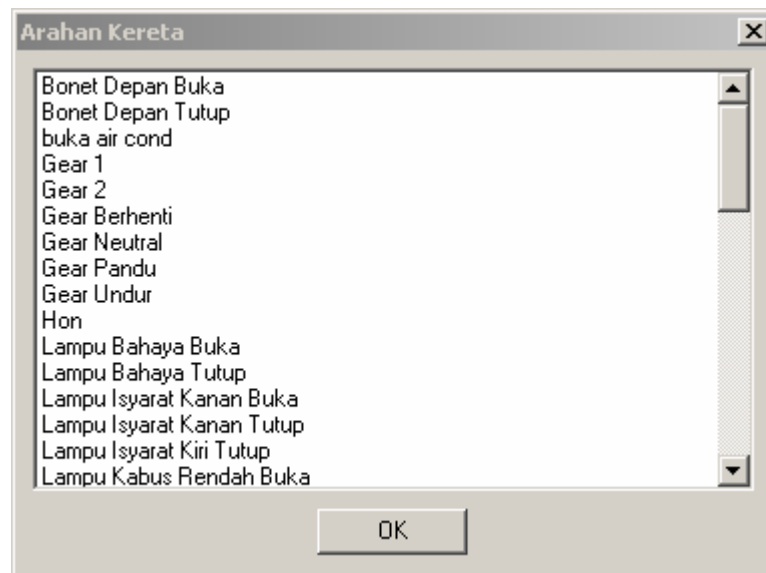
51	Radio	CD	Keluar	
52	Radio	Sebelum		
53	Radio	Selepas		
54	Radio	Kosong		
55	Radio	Satu		
56	Radio	Dua		
57	Radio	Tiga		
58	Radio	Empat		
59	Radio	Lima		
60	Radio	Enam		
61	Radio	Tujuh		
62	Radio	Lapan		
63	Radio	Sembilan		
64	Pintu	Buka		
65	Pintu	(<i>Unlock</i>) Tutup (<i>Lock</i>)		
66	Bonet	Depan	Buka	
67	Bonet	Depan	Tutup	
68	Bonet	Belakang	Buka	
69	Bonet	Belakang	Tutup	

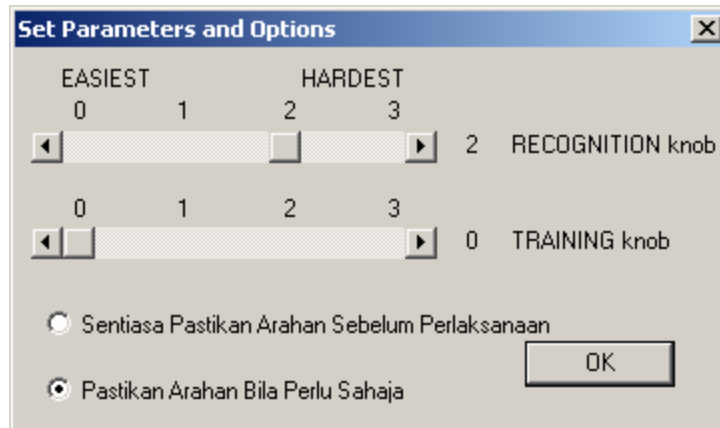
APPENDIX B: General Architecture for Speech Recognition in Microcontroller



APPENDIX C: Flow diagram chart for traditional speech recognition system

APPENDIX E: GUI for SVM-DSW





APPENDIX F: Recognition result for each iteration.

Table 3: Recognition result for each iteration.

No.	Class Posterior Probability Distribution					Recognized Word
	bonet	gear	hon	lampu	pendingin	
	pengilap	pintu	radio	tingkap		
1	0.237764	0.0438989	0.0249693	0.0888348	0.0342335	tingkap
	0.133944	0.0353765	0.043158	0.35782		
2	0.163185	0.039915	0.043469	0.1416	0.0577886	tingkap
	0.0527725	0.0418211	0.0550237	0.404425		
3	0.139863	0.0427431	0.0398873	0.220862	0.0587254	tingkap
	0.0445951	0.0509223	0.0627196	0.339683		
4	0.101942	0.0361214	0.0250104	0.410964	0.0408373	lampu
	0.0284194	0.0643569	0.0953612	0.196987		
5	0.0366623	0.0164632	0.0134021	0.705736	0.018834	lampu
	0.014567	0.0754922	0.049957	0.068886		
6	0.0126367	0.00791573	0.00587631	0.880699	0.00968108	lampu
	0.00673308	0.0379377	0.0204647	0.0180555		
7	0.0101281	0.00531958	0.00540646	0.913056	0.00793892	lampu
	0.00598565	0.0295714	0.0113987	0.0111956		
8	0.0149695	0.00790706	0.00665965	0.891389	0.016835	lampu
	0.00965993	0.0218823	0.0145904	0.0161067		
9	0.0278843	0.0129348	0.0124427	0.812891	0.0283974	lampu
	0.0218607	0.0235752	0.0252943	0.034719		
10	0.0319556	0.0281324	0.0176267	0.690126	0.0297155	lampu
	0.039901	0.0390751	0.0513724	0.0720954		
	bahaya		dalam		isyarat	
	kabus		luar		sorot	
11	0.0558017		0.0517912		0.0931264	luar

	0.255638	0.432645	0.110998	
12	0.0537842	0.0615906	0.133451	luar
	0.110894	0.534566	0.105715	
13	0.0487344	0.0758606	0.137497	luar
	0.0799268	0.565423	0.0925581	
14	0.0438174	0.0931974	0.117259	luar
	0.0604658	0.604308	0.0809519	
15	0.0391868	0.110228	0.0860941	luar
	0.0601935	0.644061	0.0602363	
16	0.0478353	0.124278	0.0827467	luar
	0.0814805	0.600273	0.063387	
17	0.0563011	0.105357	0.0725917	luar
	0.103083	0.592654	0.0700128	
18	0.0708906	0.0938672	0.0920916	luar
	0.190139	0.464178	0.0888335	
	buka		tutup	
19	0.987611		0.0123888	buka
20	0.985118		0.0148823	buka
21	0.984579		0.0154213	buka
22	0.990087		0.0126556	buka

APPENDIX G: HTK vs. DSW recognition benchmark.

Table 1: HTK vs. DSW recognition benchmark.

```

===== HTK Results =====
01F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
01G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
01H.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
01I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
01J.rec: 66.67(66.67) [H= 2, D= 0, S= 1, I= 0, N= 3]
02F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
02G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
02H.rec: 66.67(66.67) [H= 2, D= 0, S= 1, I= 0, N= 3]
02I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
02J.rec: 66.67(66.67) [H= 2, D= 0, S= 1, I= 0, N= 3]
03F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
03G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
03H.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
03I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
03J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
04F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
04G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
04H.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
04I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
04J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
05F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
05G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
05H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
05I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
05J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]

```

06F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 06G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 06H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 06I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 06J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 07F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 07G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 07H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 07I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 07J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 08F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 08G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 08H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 08I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 08J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 09F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 09G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 09H.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 09I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 09J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 10F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 10G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 10H.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 10I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 10J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 11F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 11G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
11H.rec: 66.67(66.67) [H= 2, D= 0, S= 1, I= 0, N= 3]
 11I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 11J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]

12F.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 12G.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
12H.rec: 66.67(66.67) [H= 2, D= 0, S= 1, I= 0, N= 3]
 12I.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 12J.rec: 100.00(100.00) [H= 3, D= 0, S= 0, I= 0, N= 3]
 13F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 13G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 13H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 13I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 13J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 14F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 14G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 14H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 14I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 14J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 15F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 15G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 15H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 15I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 15J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 16F.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 16G.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 16H.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 16I.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]
 16J.rec: 100.00(100.00) [H= 4, D= 0, S= 0, I= 0, N= 4]

----- Overall Results -----

SENTENCE: %Correct=93.75 [H=75, S=5, N=80]

WORD: % Correct =98.21, Accuracy=98.21 [H=275, D=0, S=5, I=0, N=280]

=====

===== DSW Results =====

01F) Lampu [0.95525] Luar [0.800069] Buka [0.949524]
 01G) Lampu [0.934712] Luar [0.713746] Buka [0.919402]
 01H) Lampu [0.907456] Luar [0.735728] Buka [0.885185]
 01I) Lampu [0.973099] Luar [0.771852] Buka [0.965305]
 01J) Lampu [0.94429] Luar [0.819392] Buka [0.923854]
 02F) Lampu [0.95233] Luar [0.754038] Tutup [0.948677]
 02G) Lampu [0.973535] Luar [0.779902] Tutup [0.974213]
 02H) Lampu [0.954505] Luar [0.809829] Tutup [0.943428]
 02I) Lampu [0.940206] Luar [0.788213] Tutup [0.951978]
 02J) Lampu [0.960676] Luar [0.782171] Tutup [0.962845]
 03F) Lampu [0.951461] Dalam [0.783957] Buka [0.940669]
 03G) Lampu [0.932895] Dalam [0.828707] Buka [0.955275]
 03H) Lampu [0.918906] Dalam [0.847175] Buka [0.934877]
 03I) Lampu [0.960721] Dalam [0.824569] Buka [0.938282]
 03J) Lampu [0.937994] Dalam [0.776698] Buka [0.946096]
 04F) Lampu [0.960685] Dalam [0.805333] Tutup [0.952942]
 04G) Lampu [0.959426] Dalam [0.836147] Tutup [0.951729]
 04H) Lampu [0.944898] Dalam [0.744647] Tutup [0.95719]
 04I) Lampu [0.950591] Dalam [0.733423] Tutup [0.947494]
 04J) Lampu [0.948294] Dalam [0.811344] Tutup [0.947427]
 05F) Lampu [0.905932] Isyarat [0.849157] Kiri [0.942797] Buka [0.956552]
 05G) Lampu [0.963499] Isyarat [0.887317] Kiri [0.947483] Buka [0.961307]
 05H) Lampu [0.951758] Isyarat [0.813338] Kiri [0.942372] Buka [0.971291]
 05I) Lampu [0.953201] Isyarat [0.833884] Kiri [0.926419] Buka [0.972484]
 05J) Lampu [0.958137] Isyarat [0.852908] Kiri [0.934799] Buka [0.967107]
 06F) Lampu [0.95647] Isyarat [0.85297] Kiri [0.948215] Tutup [0.974881]
 06G) Lampu [0.927356] Isyarat [0.844167] Kiri [0.948231] Tutup [0.973142]
 06H) Lampu [0.948919] Isyarat [0.864444] Kiri [0.943232] Tutup [0.967343]
 06I) Lampu [0.963245] Isyarat [0.861121] Kiri [0.94381] Tutup [0.969749]

06J) Lampu [0.951748] Isyarat [0.875426] Kiri [0.934034] Tutup [0.971312]
07F) Lampu [0.942825] Isyarat [0.874001] Kanan [0.916978] Buka [0.947268]
07G) Lampu [0.95116] Isyarat [0.877796] Kanan [0.924068] Buka [0.967295]
07H) Lampu [0.971695] Isyarat [0.860791] Kanan [0.927281] Buka
[0.953759]
07I) Lampu [0.963652] Isyarat [0.84805] Kanan [0.915941] Buka [0.968872]
07J) Lampu [0.954937] Isyarat [0.850175] Kanan [0.933544] Buka [0.947274]
08F) Lampu [0.968719] Isyarat [0.815345] Kanan [0.926193] Tutup
[0.949537]
08G) Lampu [0.962272] Isyarat [0.87041] Kanan [0.919358] Tutup [0.971365]
08H) Lampu [0.972211] Isyarat [0.845423] Kanan [0.765151] Tutup
[0.789228]
08D) Lampu [0.966083] Isyarat [0.859217] Kanan [0.736304] Tutup
[0.718495]
08J) Lampu [0.952086] Isyarat [0.873778] Kanan [0.91943] Tutup [0.932098]
09F) Lampu [0.973024] Bahaya [0.878628] Buka [0.961003]
09G) Lampu [0.971211] Bahaya [0.896058] Buka [0.938728]
09H) Lampu [0.956181] Bahaya [0.914243] Buka [0.969395]
09I) Lampu [0.982041] Bahaya [0.845794] Buka [0.940231]
09J) Lampu [0.978976] Bahaya [0.904151] Buka [0.959234]
10F) Lampu [0.943314] Bahaya [0.909177] Tutup [0.982075]
10G) Lampu [0.970158] Luar [0.560815] Tutup [0.700689]
10H) Lampu [0.962987] Bahaya [0.909177] Tutup [0.979983]
10I) Lampu [0.947574] Bahaya [0.912665] Tutup [0.976275]
10J) Lampu [0.957684] Bahaya [0.897574] Tutup [0.981968]
11F) Lampu [0.96898] Sorot [0.842756] Buka [0.968029]
11G) Lampu [0.925868] Sorot [0.851592] Buka [0.965276]
11H) Lampu [0.969069] Sorot [0.811795] Buka [0.894761]
11I) Lampu [0.949244] Sorot [0.815088] Buka [0.938036]
11J) Lampu [0.959954] Sorot [0.801599] Buka [0.959241]

12F) Lampu [0.94702] Sorot [0.816745] Tutup [0.963008]
12G) Lampu [0.927622] Sorot [0.679545] Tutup [0.961653]
12H) Lampu [0.936331] Sorot [0.792682] Tutup [0.969421]
12I) Lampu [0.966041] Sorot [0.786881] Tutup [0.950213]
12J) Lampu [0.961624] Sorot [0.807129] Tutup [0.947176]
13F) Lampu [0.963697] Kabus [0.789521] Rendah [0.880556] Buka [0.949271]
13G) Lampu [0.960356] Kabus [0.831689] Rendah [0.915028] Buka
[0.938877]
13H) Lampu [0.961274] Kabus [0.778452] Rendah [0.914515] Buka
[0.961052]
13I) Lampu [0.957232] Kabus [0.839224] Rendah [0.898313] Buka [0.934784]
13J) Lampu [0.967834] Kabus [0.824522] Rendah [0.91194] Buka [0.961994]
14F) Lampu [0.942174] Kabus [0.772602] Rendah [0.887414] Buka [0.874061]
14G) Lampu [0.92962] Kabus [0.873104] Rendah [0.88093] Tutup [0.943656]
14H) Lampu [0.955816] Kabus [0.855356] Rendah [0.840794] Tutup
[0.955398]
14I) Lampu [0.955704] Kabus [0.852943] Rendah [0.805477] Tutup
[0.939069]
14J) Lampu [0.953491] Kabus [0.844374] Rendah [0.861872] Tutup
[0.956002]
15F) Lampu [0.946803] Kabus [0.880016] Tinggi [0.882331] Buka [0.958131]
15G) Lampu [0.961635] Kabus [0.859668] Tinggi [0.914845] Buka [0.965659]
15H) Lampu [0.964415] Kabus [0.868469] Tinggi [0.913072] Buka [0.943313]
15I) Lampu [0.954475] Kabus [0.877031] Tinggi [0.925554] Buka [0.962971]
15J) Lampu [0.964624] Kabus [0.882529] Tinggi [0.855599] Buka [0.948843]
16F) Lampu [0.952347] Kabus [0.845449] Tinggi [0.87541] Tutup [0.942511]
16G) Lampu [0.949312] Kabus [0.830117] Tinggi [0.924919] Tutup [0.962477]
16H) Lampu [0.96457] Kabus [0.887167] Tinggi [0.885891] Tutup [0.950166]
16I) Lampu [0.955823] Kabus [0.861316] Tinggi [0.914808] Tutup [0.9608]
16J) Lampu [0.970057] Kabus [0.838947] Tinggi [0.923264] Tutup [0.950466]

----- Overall Results -----

SENTENCE: %Correct=98.75 [H=79, S=1, N=80]

WORD: %Correct=99.64, Accuracy=99.64 [H=279, D=0, S=1, I=0, N=280]

=====

Note 1: In the Overall Results, the first line gives the sentence-level accuracy based on the total number of samples. The second line is the word accuracy based on the total number of words in the whole samples. In this second line, **H** is the number of correct recognition, **D** is the number of deletions, **S** is the number of substitutions, **I** is the number of insertions and **N** is the total number of samples. The percentage number of samples correctly recognized is given by: %Correct = $H/N*100$; Accuracy = $(H-I)/N*100$.

Note 2: The transcriptions in HTK's results have been omitted and were being replaced by the labels **01-16** representing 16 sentences with 5 repetitions (**F-J** are for testing while **A-E** are for training). Bold lines in the listing constitute misrecognition in the sentence. From this table, DSW evidently outperforms HTK in speaker-dependent mode.