



UNIVERSITY of the
WESTERN CAPE

Voice-signature-based Speaker Recognition

by

Menia Al-Kilani

A thesis submitted in fulfilment of requirements for the degree of

Magister Scientiae

Faculty of Science

UNIVERSITY of the
WESTERN CAPE
Department of Computer Science

Supervisor:

Co-supervisors

Professor Isabella M. Venter

Mr Mehrdad Ghaziasgar

Mr Reg Dodds

July 2017

Abstract

Personal identification and the protection of data are important issues because of the ubiquitousness of computing and these have thus become interesting areas of research in the field of computer science. Previously people have used a variety of ways to identify an individual and protect themselves, their property and their information. This they did mostly by means of locks, passwords, smartcards and biometrics. Verifying individuals by using their physical or behavioural features is more secure than using other data such as passwords or smartcards, because everyone has unique features which distinguish him or her from others. Furthermore the biometrics of a person are difficult to imitate or steal. Biometric technologies represent a significant component of a comprehensive digital identity solution and play an important role in security. The technologies that support identification and authentication of individuals is based on either their physiological or their behavioural characteristics. Live-data, in this instance the human voice, is the topic of this research. The aim is to recognize a person's voice and to identify the user by verifying that his/her voice is the same as a record of his / her voice-signature in a systems database. To address the main research question: "*What is the best way to identify a person by his / her voice signature?*", design science research, was employed. This methodology is used to develop an artefact for solving a problem. Initially a pilot study was conducted using visual representation of voice signatures, to check if it is possible to identify speakers without using feature extraction or matching methods. Subsequently, experiments were conducted with 6300 data sets derived from Texas Instruments and the Massachusetts Institute of Technology audio database. Two methods of feature extraction and classification were considered—mel frequency cepstrum coefficient and linear prediction cepstral coefficient feature extraction—and for classification, the Support Vector Machines method was used. The three methods were compared in terms of their effectiveness and it was found that the system using the mel frequency cepstrum coefficient, for feature extraction, gave the marginally better results for speaker recognition.

Key words:

Biometrics, Speaker recognition, Feature extraction, Feature matching (classification).

Declaration

I, Menia Al-kilani, declare that *Voice signature-based speaker recognition* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted, have been indicated and acknowledged by complete references.

Signed:



Date: February 19, 2018



UNIVERSITY *of the*
WESTERN CAPE

Acknowledgments

I wish to express my sincere appreciation and gratitude to my supervisor, Professor Isabella M. Venter for her assistance, guidance and limitless encouragement. She inspired me through her passion for good research and her admirable work ethics.

In addition, special thanks to my Co-supervisors Mr Mehrdad Ghaziasgar and Mr Reginald Dodds whose insightful criticisms and valuable ideas taught me new skills.

Thanks to Prof Bagula for his helpful advice and thanks also to the staff members of the Computer Science Department for their assistance.

Thanks to all my friends as well as my classmates for their touching encouragement.

Finally, limitless thanks to my lovely family especially; my mother, Kheira, my father Mohamed, my brother, Fouad, my sisters, Madeha, Aya, Donia and Auasha, and my fiancé, Abd Elhakim, for their unconditional love, their motivation, their unrestricted financial and moral support.

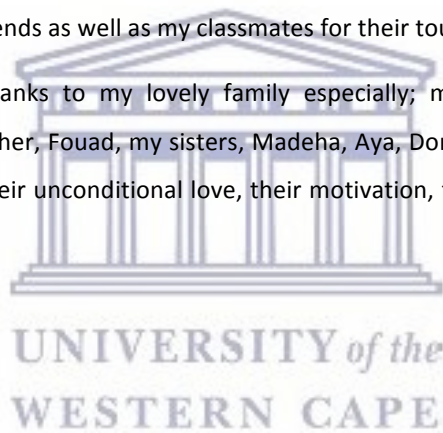


Table of Contents

Abstract -----	i
Declaration -----	ii
Acknowledgments -----	iii
Table of Contents -----	iv
List of Figures -----	vi
List of Tables -----	viii
Glossary of terms -----	ix
CHAPTER1 INTRODUCTION -----	1
1.1 <i>Background and Motivation</i> -----	1
1.2 <i>Problem Statement and Research Question</i> -----	10
1.3 <i>Research Framework</i> -----	11
1.4 <i>Contribution and Findings</i> -----	11
1.5 <i>Thesis Outline:</i> -----	12
CHAPTER 2 LITERATURE REVIEW -----	13
2.1 <i>Overview of Speaker Recognition</i> -----	13
2.2 <i>Feature Extraction</i> -----	16
2.2.1 Mel frequency cepstrum coefficient-----	17
Pre-emphasis-----	19
Framing-----	20
Hamming window-----	20
Fast Fourier transform-----	21
Mel-scale filter bank-----	22
Logarithm and discrete cosine transform-----	23
2.2.2 Linear prediction cepstral coefficients-----	23
2.2.3 Perceptual linear prediction-----	25
2.3 <i>Feature Matching (Classification)</i> -----	26
2.3.1 Support Vector Machines-----	27
2.3.2 Vector quantization-----	34
2.3.3 Gaussian mixture model-----	35
2.3.4 Artificial neural network-----	35
2.3.5 Hidden Markov model-----	35
2.3.6 Dynamic time warping-----	36
2.4 <i>Summary</i> -----	37
CHAPTER 3 RESEARCH METHODOLOGY -----	38
3.1 <i>Research Philosophy</i> -----	38
3.1.1 Epistemology-----	40
3.1.2 Theoretical perspective:-----	40

Methodology-----	41
Design science research-----	42
3.1.3 Methods-----	44
3.2 <i>Research Design</i> -----	44
3.2.1 Application of design science research-----	44
Phase 1: Visual representation cycle/phase-----	45
Data collection process-----	45
Method used-----	45
Phase 2: Mel frequency cepstrum coefficients combined with Support Vector Machine cycle/phase-----	47
Data collection process-----	47
Methods used-----	47
Phase 3: Linear prediction cepstral coefficients combined with Support Vector Machine cycle/phase-----	49
Data collection process-----	49
Methods used-----	49
3.3 <i>Summary</i> -----	49
CHAPTER 4 RESULTS-----	51
4.1 <i>Phase 1: Jaccard Similarity Method</i> -----	51
4.2 <i>Phase 2: MFCC Feature Extraction Algorithm with SVM classification.</i> -----	60
4.3 <i>Phase 3: LPCC Feature Extraction Algorithm with SVM classification.</i> -----	62
4.4 <i>Comparison of Different Methods:</i> -----	65
4.5 <i>Summary</i> -----	66
CHAPTER 5 DISCUSSION AND CONCLUSION-----	67
5.1 <i>Findings in Terms of the Research Questions</i> -----	67
5.2 <i>Difficulties Encountered</i> -----	69
5.3 <i>Future Work</i> -----	69
5.4 <i>Conclusion</i> -----	70
Bibliography -----	71
Appendices -----	77

List of Figures

<i>Number</i>	<i>Page</i>
Figure 1: Computer ownership in 2014.....	1
Figure 2: Internet access for developing countries in 2014.	2
Figure 3: Planned African undersea cables for 2015 (Song, 2015).....	3
Figure 4: Different ways of identification and authentication.	4
Figure 5: Examples of characteristics that can be used for biometric authentication	6
Figure 6: A. User identification, B. User verification/authentication.	8
Figure 7: The aim of the research.....	11
Figure 8: Speaker recognition system as adapted from (Patel & Nadurbarkar, 2015, p. 555).	14
Figure 9: Strategy of speaker recognition, A. Depicts the training stage, and B. Depicts the authentication stage.	15
Figure 10: Mel frequency cepstrum coefficient (MFCC) steps. Adapted from (Patel & Prasad, 2013, p. 35).....	18
Figure 11: The waveform after and before pre-emphasis.	19
Figure 12: Basic plot of a Hamming window. (Rachna, Singh, & Vikas, 2014, p. 274).....	21
Figure 13: Different energy distribution over frequency. (Rachna, Singh, & Vikas, 2014, p. 274).	22
Figure 14: Mel-scale filter bank, (Tallat, Shahid, Samad, & Abbasi, 2014, p. 7).	23
Figure 15: Linear prediction cepstral coefficients (LPCC) adapted from (Veton Z & Hussien A., 2015, p. 4).	25
Figure 16: Perceptual Linear Prediction (PLP), Adapted from Veton & Hussien (Veton Z & Hussien A., 2015, p. 5).....	26
Figure 17: Support Vector Machine.	29
Figure 18: Linear classification of hyperplane. (Achmed, 2014, p. 57).....	29
Figure 19: (a) Linear classification, (b) Non-Linear classification. (Achmed, 2014, p. 56)	32
Figure 20: Vector quantization codebook formation. Adapted from (Tallat, Shahid, Samad, & Abbasi, 2014, p. 9)	34

Figure 21: Hidden Markov model.....	36
Figure 22: Dynamic Time Warping (DTW). (Muda, Begam, & Elamvazuthi, 2010, p. 140) .	37
Figure 23: The four basic elements of research according to Crotty (1998).	39
Figure 24: The iterative process of a general DSR cycle consists of six basic stages (Vom Brocke & Buddendick, 2006, p. 582)	42
Figure 25: The DSR cycle applied to each of the phases.	44
Figure 26: A. Voice waveform, B. Linear frequency analysis image.	46
Figure 27: Phase 1: The first DSR cycle consisting of the visual representation phase.	51
Figure 28: The four waveform images for Speaker 1.	52
Figure 29: The waveforms of six speakers for the same phrase.	52
Figure 30: The four linear frequency analysis images of Speaker 1.	53
Figure 31: Linear frequency analysis images for six different speakers.	54
Figure 32: Six comparisons of the same speakers for all six speakers.....	56
Figure 33: Comparisons voice images of Speaker 1 with the other speakers.	57
Figure 34: Comparisons of voice images of Speaker 3 with the other speakers.....	59
Figure 35: The second DSR cycle consisting of the MFCC+SVM phase.	60
Figure 36: The accuracy of recognition for the 35 speakers using MFCC and SVM.	62
Figure 37: The third DSR cycle consisting of the LPCC+SVM phase.....	62
Figure 38: The accuracy of recognition for the 35 speakers using LPCC and SVM.	64

List of Tables

<i>Number</i>	<i>Page</i>
Table 1: The summary comparison of the biometric techniques	7
Table 2: The main results obtained using MFCC combined with SVM.....	60
Table 3: The main results obtained using LPCC combined with SVM.	63
Table 4: Comparison between MFCC+SVM and LPCC+SVM.	65
Table 5: Statistical comparison of the two methods	65



UNIVERSITY *of the*
WESTERN CAPE

Glossary of terms

ACC	Average of accuracy
ANN	Artificial neural network
ASR	Automatic speaker recognition
DCT	Discrete cosine transform
DNA	Deoxyribonucleic acid
DSR	Design science research
DTW	Dynamic time warping
EM	Expectation-maximization algorithm
F₀	Fundamental Frequency
FAR	False acceptance rate
FFT	Fast Fourier transform
FRR	False rejection rate
GMM	Gaussian mixture model
HMM	Hidden Markov model
HTK	Hidden Markov model toolkit
IDFT	Inverse discrete Fourier transform
LBG	Linde, Buzo and Garay algorithm
LP	Linear prediction
LPC	Linear predictive coefficient
LPCC	Linear prediction Cepstral coefficient

LSVM	Linear support vector machine
LDB	Local discriminant bases
MFCC	Mel frequency cepstrum coefficient
PLP	Perceptual linear prediction
RBF	Radial basis function
SRS	Speaker recognition system
SVM	Support Vector Machines
TIMIT	Texas instruments and Massachusetts Institute of Technology
UWC	University of the Western Cape
VQ	Vector quantization



Chapter1 Introduction

1.1 Background and Motivation

Security has become an important issue in modern society because of our interconnectedness, and the increasing use of the Internet. Figure 1, shows computer ownership in 2014 and Figure 2 shows Internet access in the US and developing countries around the world in the same year. This technology usage and interconnectedness allows crimes like hacking, phishing to flourish.

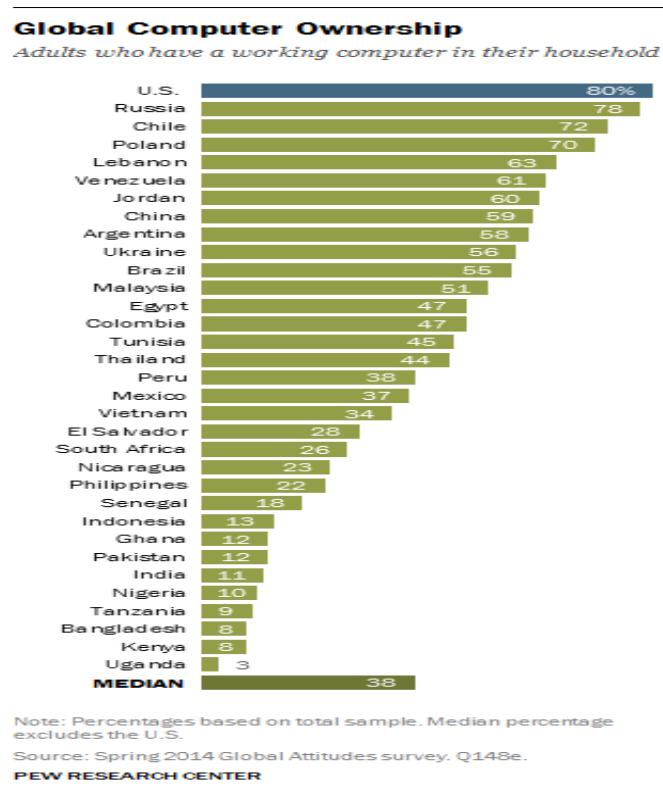
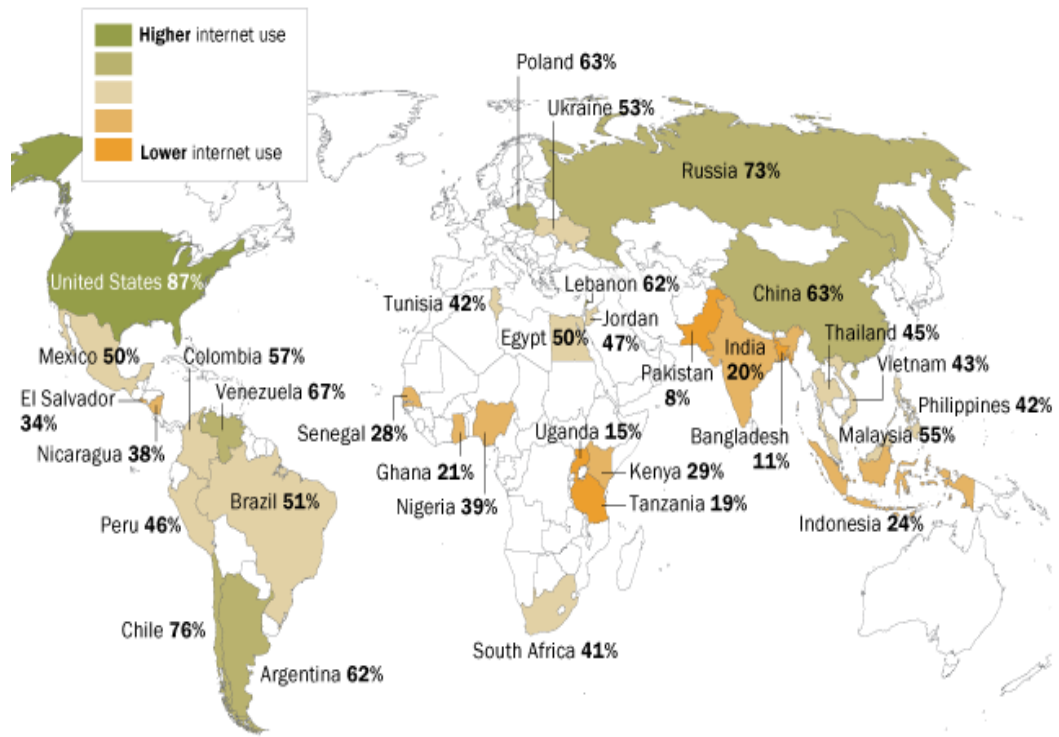


Figure 1: Computer ownership in 2014¹.

¹ <http://www.pewglobal.org/2015/03/19/internet-seen-as-positive-influence-on-education-but-negative-influence-on-morality-in-emerging-and-developing-nations/technology-report-16/>



Note: U.S. data comes from a January 2014 Pew Research Center survey.
 Source: Spring 2014 Global Attitudes survey, Q67 & Q69.

PEW RESEARCH CENTER

Figure 2: Internet access for developing countries in 2014².

² <http://www.pewglobal.org/2015/03/19/internet-seen-as-positive-influence-on-education-but-negative-influence-on-morality-in-emerging-and-developing-nations/technology-report-24-2/>

As can be seen in Figure 2, the so-called third-world countries, such as countries on the African continent were not as interconnected as, for example the United States of America, in 2014. However, as is shown in Figure 3, the African continent is fast becoming more connected and thus, there is a pressing need in these countries to become more aware of security. Security breaches can negatively affect individuals: for example, breaches can result in the loss of a person’s identity and the control of that person’s personal data. Security breaches can present a serious risk to organizations and businesses. As hackers get smarter, security needs to become more complex.

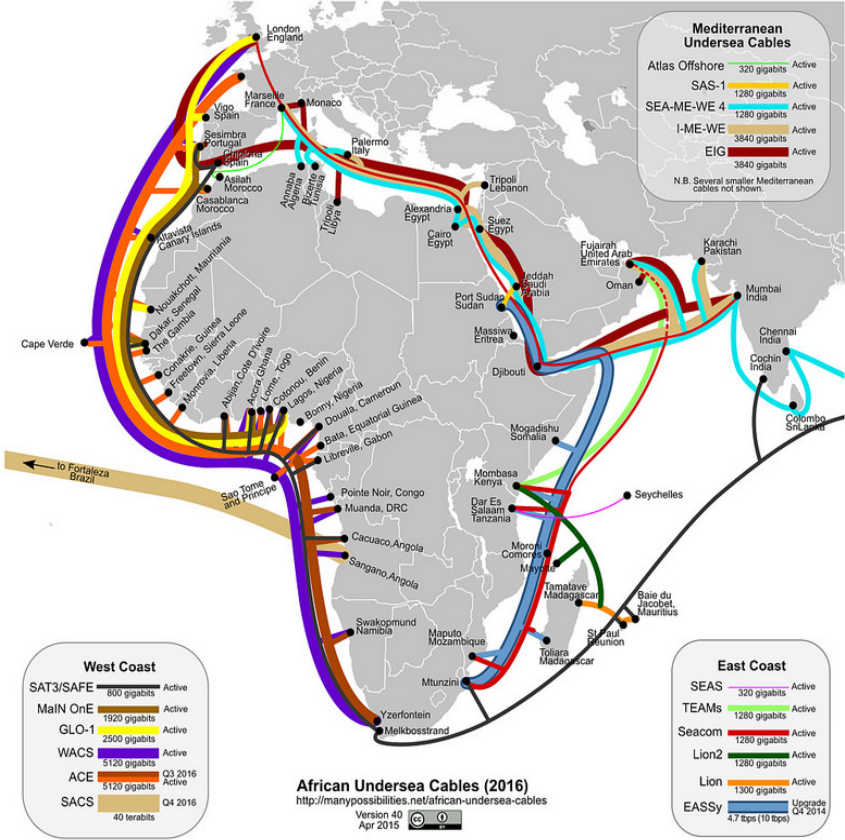


Figure 3: Planned African undersea cables for 2015 (Song, 2015).

The recognition of a person has wide applications, which can vary from identifying a person to give them permission to enter a home or an office, to access his / her computer, etc. to

the secure identification of a person at a border post. Representations of identity, such as identity cards and passwords, are not a completely secure and reliable identification method because they can easily be tampered with, misplaced, shared, or stolen (Jain, Nandkumar, & Nagar, 2008). Currently there is an increased interest in providing higher levels of security and protection for individuals and communities.

The authentication of a person's identity is a challenging problem. Currently three common ways are used for authentication, that are to establish credibility and in general, for identification. Most of these commonly used methods are based on:

- What devices a person has—for example physical keys or smartcards.
- What a person knows—such as passwords, personal information or PINs.
- The characteristics of the person—such as biometric modalities. See Figure 4.

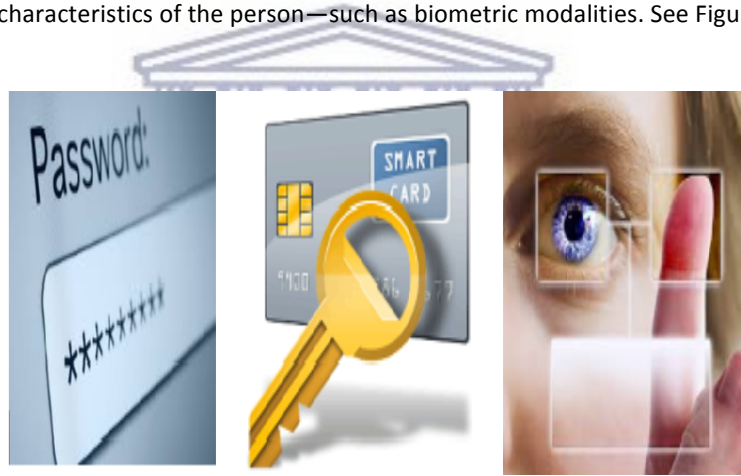


Figure 4: Different ways of identification and authentication.

When considering these three authentication methods to identify a person, it can be seen that the last mentioned method, the biometric characteristics of a person, which uses the individual's physical or behavioural features for authentication,—ensure a more secure identification than using passwords or smartcards, because everyone has unique features, which distinguish him / her from others. According to Montrose *et al.* biometric-based security systems are nearly impossible to defraud (Monrose, Fabian, & Aviel, 2000) (Pal, Pal, & Blumenstein, 2014) (Prakash & Gupta, 2015).

Biometrics have existed for centuries, where parts of bodies and aspects of behaviour have been used historically, in an unsophisticated manner, as a means to check a person's identity. For example: in ancient Egypt unique body traits were used to identify construction labourers in order to ensure the fair distribution of food; and fingerprint studies date back to ancient China. A person's face or sound of his / her voice was most often used to identify a person; signatures have been the de facto method of authentication in banking and for legal contracts, etc. for many years. Automated biometrics, however, have only been in use for the last 40 years (Boukhonine, Krotov, & Rupert, 2005).

The word *biometric* is derived from two Greek words "bios" and "metra", bio means life and metra means measure (Pal, Pal, & Blumenstein, 2014). However, several scientific definitions for biometrics exist:

- Jain *et al.* define biometrics as the measurement and statistical analysis of an unchanging biological characteristic (Jain, Hong, & Pankanti, 2000) but re-define it as the science of using the physical, chemical or behavioural features of individuals to establish a person's identity—thus the metrics related to a human characteristic (Jain, Flynn, & Ross, 2007);
- According to Prabhakar *et al.* biometrics links a physical or behavioural attribute of a person, to a person's identity, by using specific methods that focus upon individual differences. These differences, as depicted in Biometric characteristics can be divided into two main classes: physiological and behavioural.
- Physiological characteristics depend on the features of a person's body, such as: fingerprints, facial features, hand shape, iris features, etc.
- Behavioural characteristics are related to the style of a person's behaviour such as their manner of speaking, their hand signature or writing style, and their walking style (Bragagnini, Della Luna, Nocentini, & Turolla, 2014) (Prabhakar, Pankanti, & Jain, 2003). See Figure 5.

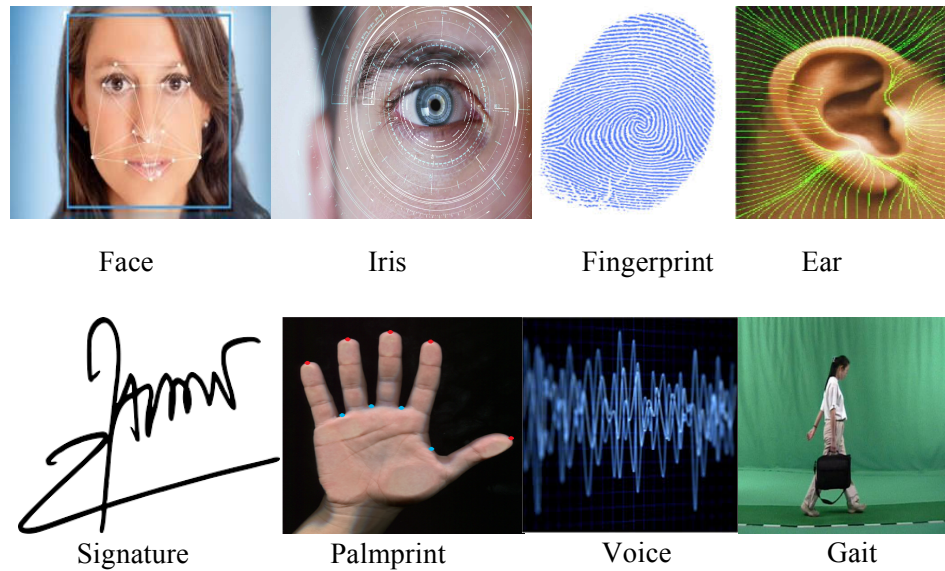


Figure 5: Examples of characteristics that can be used for biometric authentication

Biometrics have desirable characteristics with respect to authentication, namely:

- Reliability and distinctiveness: no two individuals share the same biometric characteristics. They are unique to each individual.
- Permanence: this characteristic does not change over time.
- Collectibility: the characteristic can be measured quantitatively.
- Universality: individuals have unique characteristics or features that will not change even if they are involved in an accident or succumb to a disease.
- Acceptability: their use should be acceptable in general society.
- Circumvention: they should be difficult or impossible to imitate or forge (Prakash & Gupta, 2015).

These characteristics have led to the widespread dissemination of authentication systems, generally using one or more of the following: fingerprints, images of the face, iris recognition, handwriting recognition, signatures, hand geometry, voice prints, palm prints or a person's gait (Jain, Nandkumar, & Nagar, 2008) (Mordini & Tzovaras, 2012) (Prabhakar, Pankanti, & Jain, 2003)

Table 1: The summary comparison of the biometric techniques

Biometric Identifier	Universality	Distinctiveness	Permanence	Collectability	Performance	Acceptability	Circumvention	Total
DNA	H	H	H	L	H	L	L	4H, 3L
Ear	H	M	H	M	M	H	M	3H, 4M
Face	H	L	M	H	L	H	H	4H, 1M, 2L
Facial thermo gram	H	H	L	H	M	H	L	4H, 1M, 2L
Fingerprint	M	H	H	M	H	M	M	3H, 4M
Gait	M	L	L	H	L	H	M	2H, 2M, 3L
Hand geometry	M	M	M	H	M	M	M	1H, 6M
Hand vein	M	M	M	M	M	M	L	6M, 1L
Iris	H	H	H	M	H	L	L	4H, 1M, 2L
Keystroke	L	L	L	M	L	M	M	3M, 4L
Odor	H	H	H	L	L	M	L	3H, 1M, 3L
Palm print	H	H	M	L	H	L	L	3H, 1M, 3L
Retina	H	H	M	L	H	L	L	3H, 1M, 3L
Signature	L	L	L	H	L	H	H	3H, 4L
Voice	M	L	L	M	L	H	H	2H, 2M, 3L

The summary of Jain *et al.*'s comparison of the biometric techniques is listed in Table 1 where H means high reliability, M means medium and L means low. As can be seen iris and face are best, but there are many disadvantages of using these biometrics. For example, in case of using the iris, the iris scanners may be tricked easily through the use of a high-quality image of the iris or face rather than the real thing. Iris scanners cannot be used for individuals with diabetes or some other serious diseases that may cause changes in the iris. In addition the iris scanners are often difficult to adjust and they are more expensive compared to other biometric measuring devices (Saini & Rana, 2014).

Using the face for recognition is not effective if the lighting is bad or weak, if hair is long, or other objects partly cover the subject's face. In addition using this type of biometric is more expensive and complex than many others. However, for all the biometrics except voice, the person needs to be present (Saini & Rana, 2014).

Biometric systems are pattern-recognition systems. These systems typically work in one of two modes: verification or identification:

1. In verification mode the system compares the captured biometric characteristic of the user with their biometric template, which has been previously saved in the system's database. The question being answered is "is this Peter?" or "is this Peter's voice?"

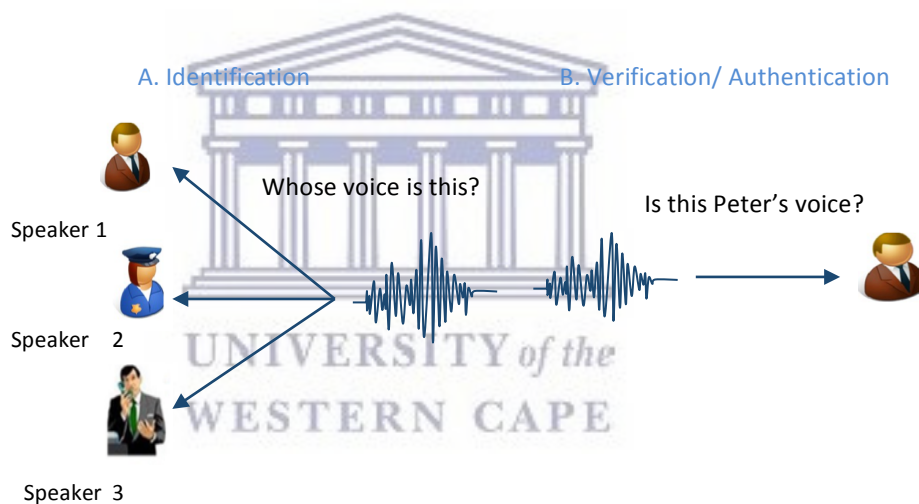


Figure 6: A. User identification, B. User verification/authentication.

2. While in identification mode the question being answered is "Who is this person?" or "Whose voice is this?" to identify the individual the entire template database is searched for a match. The system does a comparison of one to many to establish an individual's identity (Prabhakar, Pankanti, & Jain, 2003) as can be seen in Figure 6.

Throughout this thesis the generic term "recognition" is used for both verification and identification.

It is interesting to note that behavioural biometrics and new advanced sensor technologies, which use and analyse the various bodily behavioural characteristics for recognition, have been increasing. Furthermore, new sensors and networks have been introduced in intelligent environments, which are capable of detecting physical movement and motion-based characteristics, communication properties, and the existence and the incorporation of these characteristics over time and space. According to Mordini & Tzovaras this is very promising for biometrical recognition (Mordini & Tzovaras, 2012).

The voice, a behavioural biometric, when compared to other biometrics, is, according to Pal *et al.* the most difficult to imitate. It is even possible to distinguish between voice signatures of identical twins, who share the same DNA (deoxyribonucleic acid), which makes this technique sufficiently reliable (Pal, Pal, & Blumenstein, 2014).

Most speaker recognition systems use the voice signal's features to discriminate between individuals; these features can vary greatly from one speaker to another depending upon their anatomy and behavioural characteristics.

Speaker recognition can be used in many applications for example:

- for access control such as to access computer networks or websites or to access physical facilities;
- to authorise transaction such as telephone banking or remote credit card purchases;
- in forensics for law enforcement;
- for management using speech data such as voice mail browsing or speech skimming;
- for personalisation such as smart answering machines or voice customization (Kinnunen & Li, 2010) (Mishra, 2012).

To extract the unique characteristics from a voice signal is the basic objective of speaker recognition. This can be achieved by converting an uttered phrase from analogue to digital format; the digital format permits the feature extraction methods to identify unique vocal characteristics for creating a voiceprint or a voice signature. Voice recognition feature extraction methods are described in detail in the following chapter.

Speaker recognition is subdivided into: text-dependent speaker recognition and text-independent speaker recognition (Patel & Prasad, 2013). In text-dependent recognition the system has previous knowledge of the text spoken by the speaker. The speaker reads a set of specific words, that knowledge of spoken text enhances system performance. It is used for strong control over user input applications, while in text-independent recognition there are no restrictions on spoken text, the speakers are allowed to use any words. The system must be able to recognise the user from any text. This type of recognition is used when less control is needed over user input applications, and it is the more challenging of the two tasks (Salna & Kamaraukas, 2015) (Xu, 2015) (Reynolds, 2002).

1.2 Problem Statement and Research Question

The aim of this investigation is to identify users by verifying that their voice matches the recorded voice signature. This investigation will concentrate on text-dependent speaker recognition.

The main research question thus is: “What is the best way to identify a person using his / her voice signature?”

This research question can be further subdivided into the following:

- What feature extraction methods should be used?
- What matching or classification method would be effective?
- What combinations of methods are reliable to authenticate a person’s voice signature?

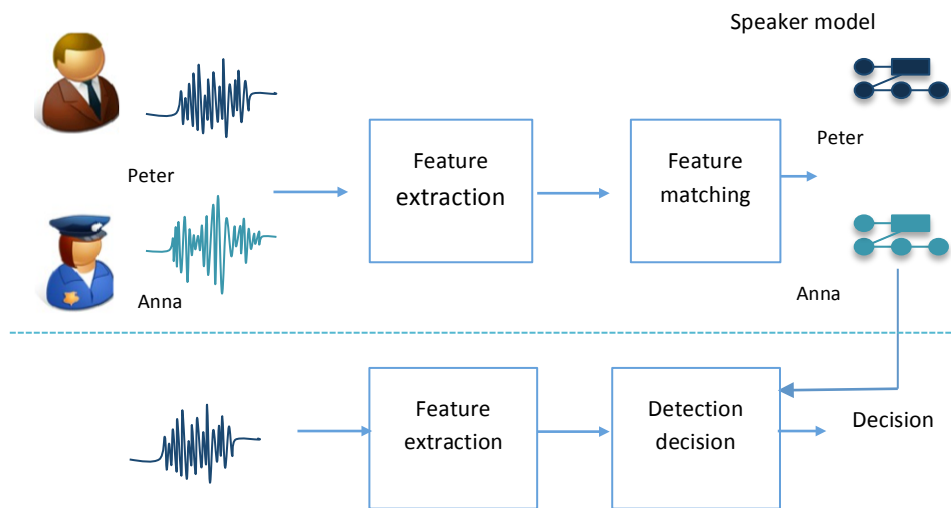


Figure 7: The aim of the research.

1.3 Research Framework

A research methodology is the overall approach or design behind the choice of specific methods, which in turn links the choice of methods to the desired outcomes. Methods refer to the techniques used to gather and analyse data with respect to a specific research question or hypothesis (Crotty, 1998).

In this research, design science research (DSR) was used, more specifically the research methodology of Vom Brocke and Buddendick (2006), was used, where artefacts are developed to solve a problem and the knowledge derived during the process, is recorded and contributes to the field of knowledge (Vom Brocke & Buddendick, 2006). The methods used with this methodology were quantitative and the artefact, a conceptual voice recognition system.

1.4 Contribution and Findings

Initially visual representations of voice images created with Audacity³ were compared. Subsequently two feature extraction algorithms were considered—mel frequency cepstrum coefficient (MFCC) and linear prediction cepstral coefficient (LPCC)—for feature

³<http://www.audacityteam.org/>

extraction, and a Support Vector Machines (SVM) algorithm was considered for classification.

The MFCC is used for feature extraction and SVM is used for classification with *text-dependent* speaker recognition. The MFCC feature extraction algorithm will be shown to give the slightly higher accuracy, when compared with the other two methods. The LPCC feature extraction method gives results similar to the MFCC with slightly less accuracy. It will be shown that the visual representation method gives the poorest results.

1.5 Thesis Outline:

In this chapter an overview of the research is presented. It includes the background and motivation, the aim of the investigation, the posing of the research questions and the research design. The rest of the thesis is laid out as follows:

In **Chapter 2**, the literature that deals with the identification of voice signatures is reviewed.

In **Chapter 3** the research design is discussed and the philosophical foundations upon which the research is based, is explained. The theoretical perspective and the chosen methodology is described. The methods chosen for collecting and analysing the data is motivated and explained.

Chapter 4 presents the results of the various experiments and in

Chapter 5, the research findings are analysed and discussed and some recommendations for future work are made.

Chapter 2 Literature Review

In the previous chapter an overview of the research was presented. In this chapter, existing literature that relates to voice signature recognition is discussed. The key concepts for speaker recognition are overviewed. These concepts include voice recognition, feature extraction, and feature classification.

The chapter is divided into three parts: Section 2.1 gives an overview of speaker recognition in general; in Section 2.2 the different algorithms that can be used to extract a feature, which is an important aspect of speaker recognition, are given; and in Section 2.3 an overview of the different methods which can be used for matching or classification of voice signatures, is given.

2.1 Overview of Speaker Recognition

Speech is the most natural way humans communicate. Human speech contains numerous discriminative features, which can be used to recognize a speaker by his voice. According to Kinnunen, the automatic processing of a specific person's voice using the characteristics of his / her voice is called speaker recognition (Kinnunen & Li, 2010). Whereas the processing of a speaker's voice to identify what the speaker is saying, is called speech recognition (Beigi, 2011). Speaker recognition includes individual identification, authentication, and classification, as well as segmentation, tracking and detection of speakers. Speaker recognition is a general term used to describe any procedure, which involves the identification of users based on their voice characteristics (Lee, Soong, Paliwal, & (Eds), 2012). The voice is a unique characteristic of each person. No two individuals sound identical, because of the difference in vocal tract shapes, larynx sizes, and other parts of their voice production organs (Xu, 2015).

A voice can be presented graphically by describing a depiction of the pattern of sound pressure variation in the time domain or the frequencies of sound per second. This graphical form is called a voice waveform.⁴ These waveforms are different for each person

⁴ www.digitizationguidelines.gov/term.php?term=waveformsound

because of the difference in sound pressure variation which depend on vocal tract shapes and larynx sizes (Xu, 2015). In addition; the difference in the frequency range also has an influence (Latinus & Belin, 2011). Thus a sound waveform might be useful for recognising a speaker.

Voice waveforms can be obtained easily using computer software such as Audacity, Wave pad and Adobe Audition. These programs give the user a visual impression of what has been recorded by viewing the waveform (Christensson, 2006).

A set of human voice features, characteristics, or metrics can be used for the recognition process, these are: *pitch, frequency, tone* and *cadence* (Mansour, Salh, & Mohammed, 2015). One of the most important characteristics of the human voice is the frequency of the human voice, which has a range of up to 5 KHz (Patel & Prasad, 2013).

For recognising the individuals from their voice, it is necessary to convert the speaker's voice into a digital signal with data representing each level of the voice signal in discrete time steps, and then using feature extraction to produce voice features (Muda, Begam, & Elamvazuthi, 2010).

According to the recognition system framework shown in Figure 8, speaker recognition is divided into two main components: feature extraction and feature classification or matching (Patel & Nadurbarkar, 2015).

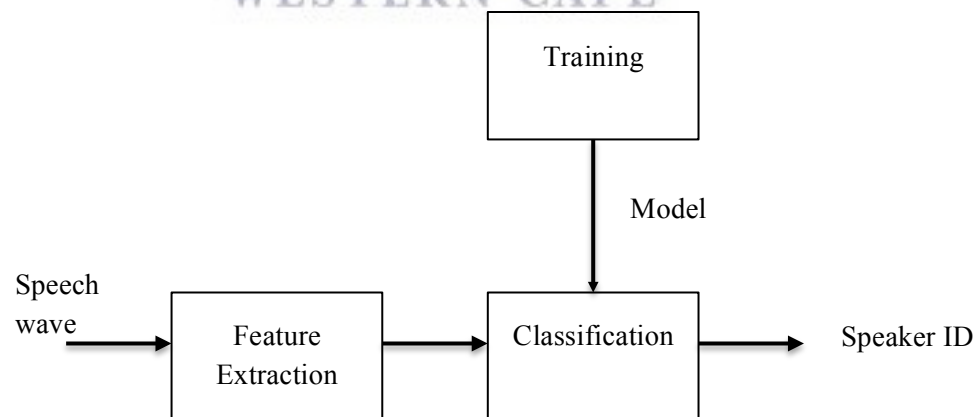


Figure 8: Speaker recognition system as adapted from (Patel & Nadurbarkar, 2015, p. 555).

The strategy for text-dependent and text-independent speaker recognition comprises two stages:

1. The training stage.

For this stage the speech signal is taken from an unknown speaker and feature vectors from the speech signal are extracted. These can then be used to identify the unknown speaker. By using these features a speech model or voiceprint is built. The basic objective of this stage is to identify the unknown speaker from a set of known speakers. See A in Figure 9.

2. The recognition, authentication or speaker classification stage.

For this stage the unknown speaker needs to be identified. A speech model or a voiceprint, which was built in the previous stage is used for identification, if the match is above a predefined threshold the identity is accepted (Salna & Kamaraukas, 2015). See B in Figure 9.

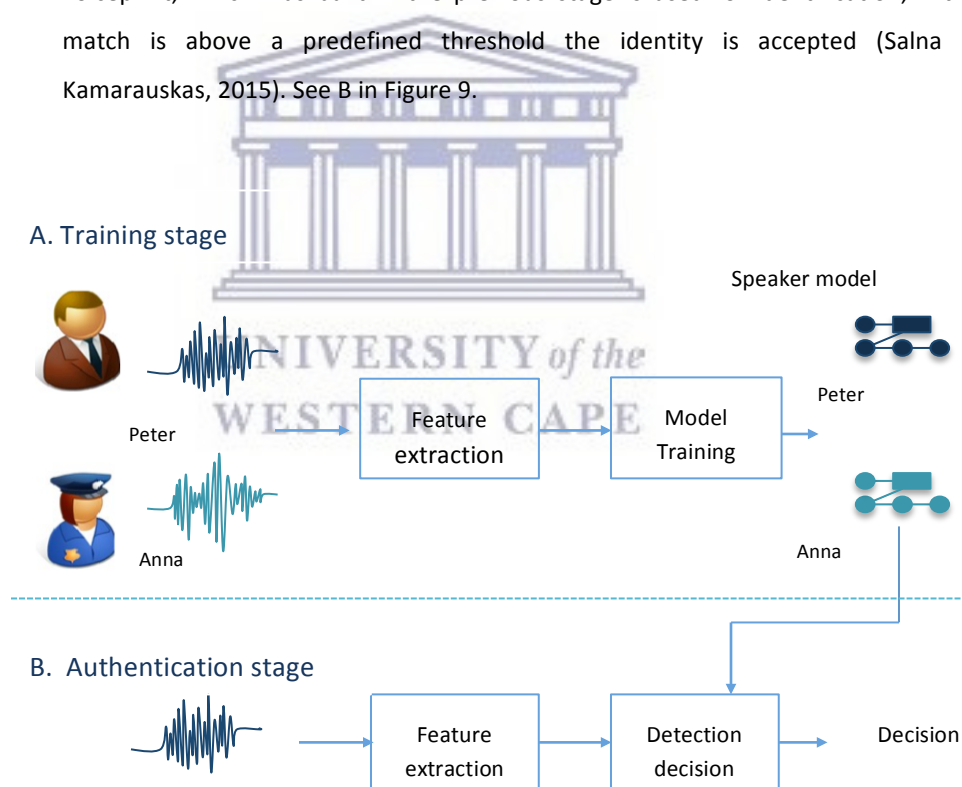


Figure 9: Strategy of speaker recognition, A. Depicts the training stage, and B. Depicts the authentication stage.

2.2 Feature Extraction

Speaker recognition is divided into two components. In this section the first component, feature extraction, will be explained.

Feature extraction is the first and an important step in speaker recognition systems, according to Patel & Prasad (2013). The objective of this step is to identify the components of the acoustic signal or to convert the speech waveform to a set of features or parameters, i.e., decreasing the number of resources required to describe the data. Depending on the specific system used to obtain a good representation of speaker features, appropriate information is estimated in a suitable form and size (Patel & Nadurbarkar, 2015).

According to Vladutescu, the human voice has nine features, namely: fundamental frequency (F_0), vocal register, tone (height, intonation), volume, accent, diction, timbre of phonation, average frequency and verbal flow (Vladutescu S. , 2013). These features or characteristics of the human voice have been considered by researchers such as Knapp, who underlined three behavioural characteristics of the human voice namely: rate, pitch and intensity (Knapp, 2008). While Poyatos considered: timbre, resonance, intensity or volume, tempo, pitch (level, intervals, range) (Poyatos, 2002). Pitch, intonation, emphasis, volume, rhythm, timing, and tempo were identified by Glenn and Meservy whereas Burgoon hold that there are three defining features of the voice: pitch, intonation, and speaking tempo (Vladutescu S. , 2013).

Only two of the nine identified characteristics, frequency, and vocal register are fixed and unchangeable for each individual. Frequency is quantitative and a constant similar to a fingerprint. The other seven characteristics are qualitative (Vladutescu S. , 2013). For speaker recognition, frequency is the main feature, which is extracted using a feature extraction method (Mansour, Salh, & Mohammed, 2015).

As a rule the value of the average frequency of an individual's voice is mainly a function of the size of the vocal folds, where in general, males have vocal folds larger than females. As a result of this, the frequency values and the pitch of male voices are lower than frequency values and the pitch of female voices (Latinus & Belin, 2011). The values are between 85Hz

and 180Hz for males and between 165Hz and 255Hz for females (Vladutescu S. , 2013). The voice pitch is the rate of vibration of the vocal folds.

Since an audio signal changes constantly, it is assumed that when a sample is taken over a short time scale, a period of between 5 and 100 ms, the audio signal will not change much. Thus statistically the audio signal can be considered to be stationary *or quasi-stationary*. However if a sample of the signal is taken over a longer period of time, in the order 200 ms or more, the characteristics of the signal change to reflect the different speech sounds being spoken. Hence, the most common way to characterize an acoustic signal is to do a short-time spectral analysis. A wide range of possibilities then exists for parametrically representing the acoustic signal for speaker recognition. For example the mel frequency cepstrum coefficient MFCC, linear prediction cepstral coefficient LPCC, perceptual linear prediction (PLP) (Hasan, Jamil, & Rahman, 2004) and local discriminant bases (LDB) (Tiwari, 2010) are all methods that can be used to do short time spectral analysis. The cepstrum is a sequence of numbers that characterizes a frame of speech. The algorithms that can be used will be explained in more detail in Sections 2.2.1—2.2.3.

2.2.1 Mel frequency cepstrum coefficient

To recap, *speech recognition* distinguishes *what* the speaker said but *speaker recognition* identifies *who* the speaker is. Perhaps the most commonly used acoustic parameters for speaker voice recognition are MFCCs. MFCCs are widely used in automatic speech or speaker recognition (ASR), because of the simplicity of the procedure for implementing MFCCs (Chakraborty, Talele, & Upadhya, 2014), and also because they are efficient (Tiwari, 2010). In addition they are considered as the best available approximation of the human ear (Gulzar, Singh, & Sharma, 2014). MFCCs were developed by Davis and Mermelstein more than three decades ago (Davis & Mermelstein, 1980) (Saleh, Ibrahim, & Ramli, 2014). They are a result of a cosine transform of the logarithm of the short-term energy spectrum expressed on a mel-frequency scale (Tiwari, 2010).

MFCCs are based on the perceptual characteristics of the human auditory system, which cannot perceive frequencies over 1Khz (Bharti & Bansal, 2015). Human perception does not follow a linear scale, the subjective pitch is measured on a scale called the 'mel scale', and

each tone frequency is measured in Hz. The mel-frequency scale uses a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1kHz (Muda, Begam, & Elamvazuthi, 2010), (Huang, Acero, Hon, & Foreword By-Reddy, 2001) and (Murty & Yegnanarayana, 2006).

The block diagram of the structure of an MFCC processor is shown in Figure 10, and includes the following steps:

- Pre-Emphasis.
- Framing.
- Hamming Window (Windowing).
- Fast Fourier Transform (FFT).
- Mel-scale filter bank.
- Logarithm and Discrete Cosine Transform (DCT).

Each step of the procedure will be explained in more detail in the next six sections

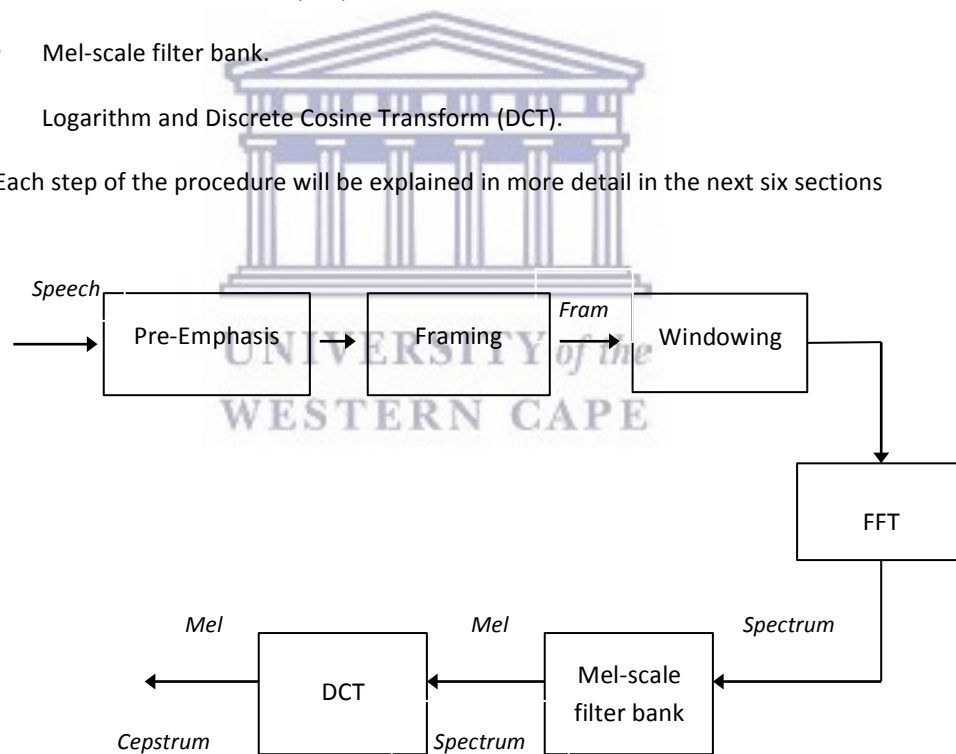


Figure 10: Mel frequency cepstrum coefficient (MFCC) steps. Adapted from (Patel & Prasad, 2013, p. 35).

Pre-emphasis

During this processing higher frequencies are emphasised by passing the voice signal through a filter. This step compensates for the pent-up part of the signal during sound production and improves or increases the energy of the signal at high frequency in order to get better results.

If $x(n)$ is a sound signal then

$$x_2(n) = x(n) - \alpha \times x(n - 1). \quad (1)$$

Where $x_2(n)$ is the output of the filter and its normalization factor varies between 0.9 to 1.0 (Rachna, Singh, & Vikas, 2014) (Chakraborty, Talele, & Upadhyay, 2014).

For example if $\alpha = 0.95$

$$x_2(n) = x(n) - 0.95 \times x(n - 1). \quad (2)$$

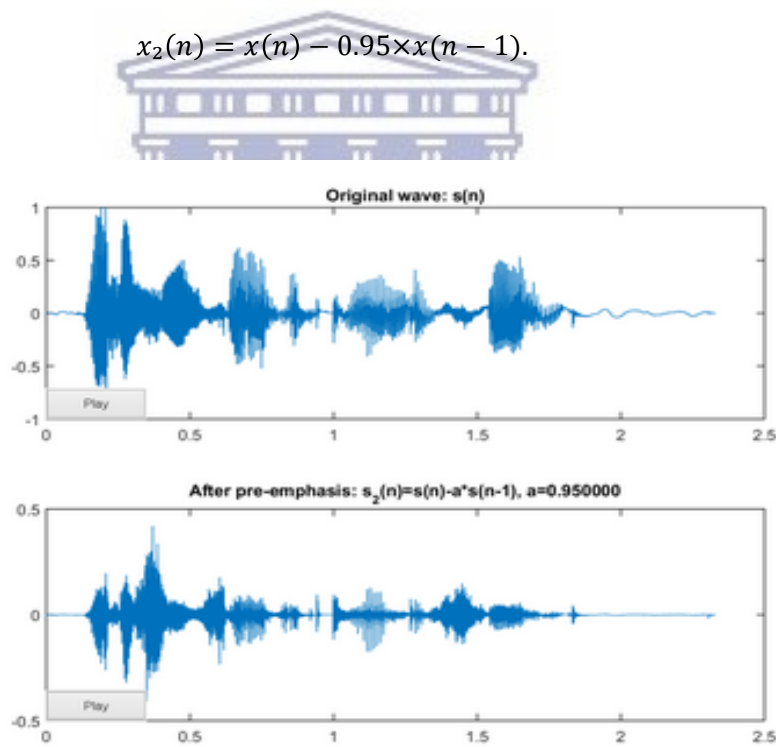


Figure 11: The waveform after and before pre-emphasis.

It assumes that 95% of any one sample originates from a previous sample. Figure 11 shows the difference between the original waveform and the waveform after pre-emphasis processing (Chapaneri, 2012).

Framing

Framing is very important to ensure good results, especially where variation of amplitude is great, such as in a large signal compared to a small signal. For this step the speech samples are converted from an analogue signal to a digital signal, called analogue to digital conversion or ADC. This is then segmented into small frames with lengths of 20–40 ms, with an overlap of 1/3– 1/5 of the frame size. The sound signal is cut into frames of N samples, adjacent frames are detached by M ($M < N$) typical values used are $M=100$ and $N=256$. If the sample rate is 16 kHz and frame size is 256, then the frame duration is $256/16000s = 0.016$ sec or 16 ms. In addition for a 1/3–1/2 overlap, there will be an overlap of 128 points, and then the frame rate is $16000/(256-128) = 125$ frames per second. Overlapping is used to produce continuity within frames (Jain & Sharma, 2013).

Hamming window

The window is used to integrate all the closest frequency lines. To keep the continuity of the first and last point of the frame, all frames will be multiplied with the hamming window (Jain & Sharma, 2013) (Joshi & Zalte, 2013). See Figure 12.

The hamming window is defined as $W(n)$ where:

N = number of the samples in each frame.

$Y(n)$ = output signal.

$X(n)$ = input signal.

$$Y(n) = X(n) \times W(n), \quad (3)$$

$$W(n) = 0.54 - 0.46 \times \cos\left(\frac{2\pi n}{N-1}\right), \quad (4)$$

where $0 \leq n \leq N-1$.

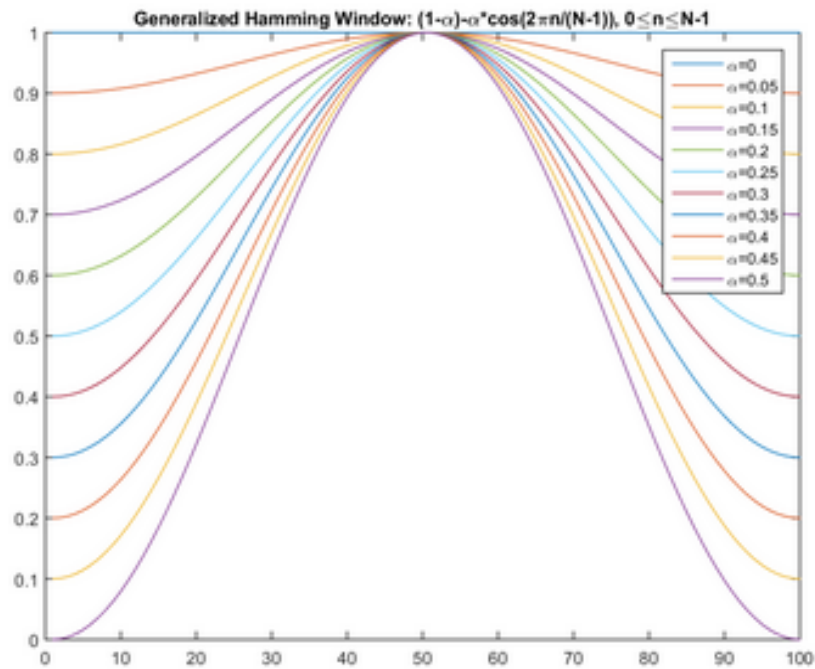


Figure 12: Basic plot of a Hamming window. (Rachna, Singh, & Vikas, 2014, p. 274).

Fast Fourier transform

The fast Fourier transform (FFT) converts each frame from the time domain into the frequency domain (Swaminathan & Jayasankar, 2015). FFT is calculated for each frame to extract the signal frequency components in the time domain. FFT is used to accelerate the processing.

The logarithmic mel-scaled filter bank is applied to the Fourier transformed frame. This scale is approximately linear up to 1 kHz, and logarithmic at greater frequencies. The equation below establishes the relation between the frequency of a signal and the mel scale (Dave, 2013)

$$f(mel) = [2595 \times \log(1 + f) / 700] \quad (5)$$

FFT is usually performed to obtain the volume frequency response of each frame, since spectral analysis displays the different timbres in voice signals matched by different energy distributions over frequencies (Rachna, Singh, & Vikas, 2014). Figure 13 depicts the energy spectrum using FFT for the original signal and for the windowed signal.

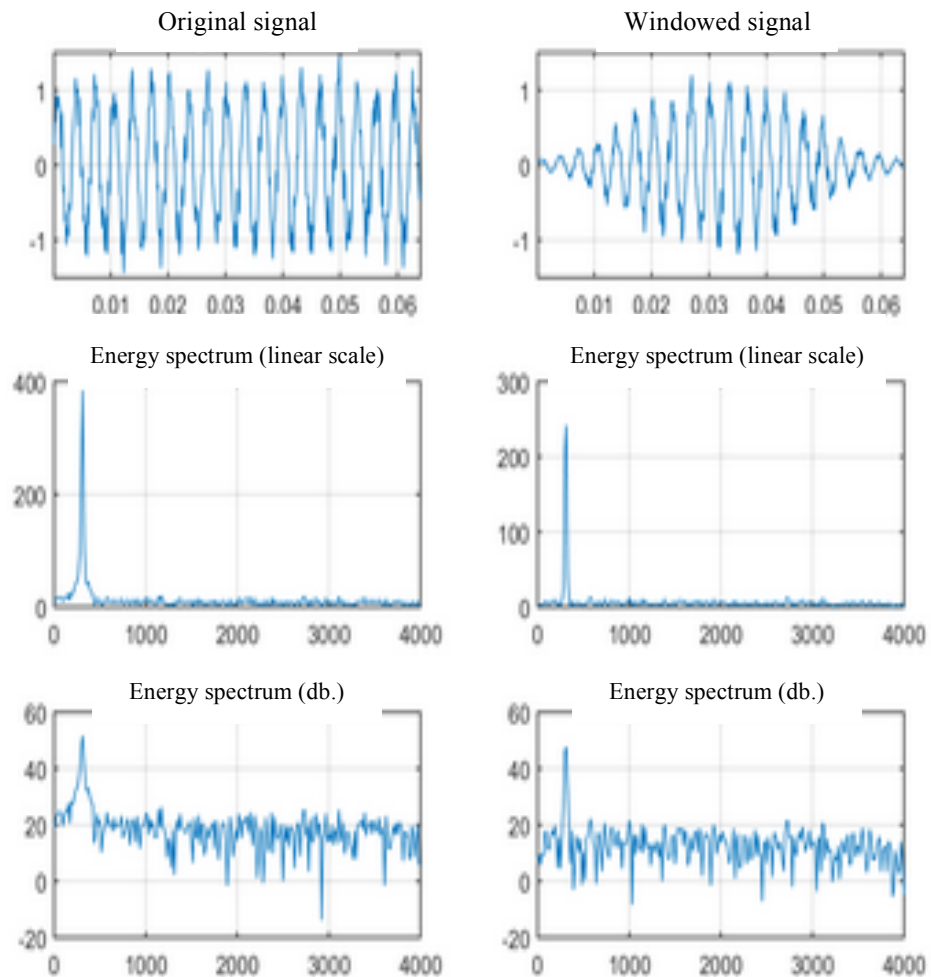


Figure 13: Different energy distribution over frequency. (Rachna, Singh, & Vikas, 2014, p. 274).

Mel-scale filter bank

The mel scale is a scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz tone, 40 dB above the listener's threshold, with a pitch of 1000 mels.

The mel-scale filter bank is a set of triangular filters that are used to compute a weighted sum of filter spectral components. The frequency range of the spectrum is very wide and

does not follow a linear scale, so it is approximated by a mel scale (Janse, Magre, Kurzekar, & Deshmukh, 2014). See Figure 14.

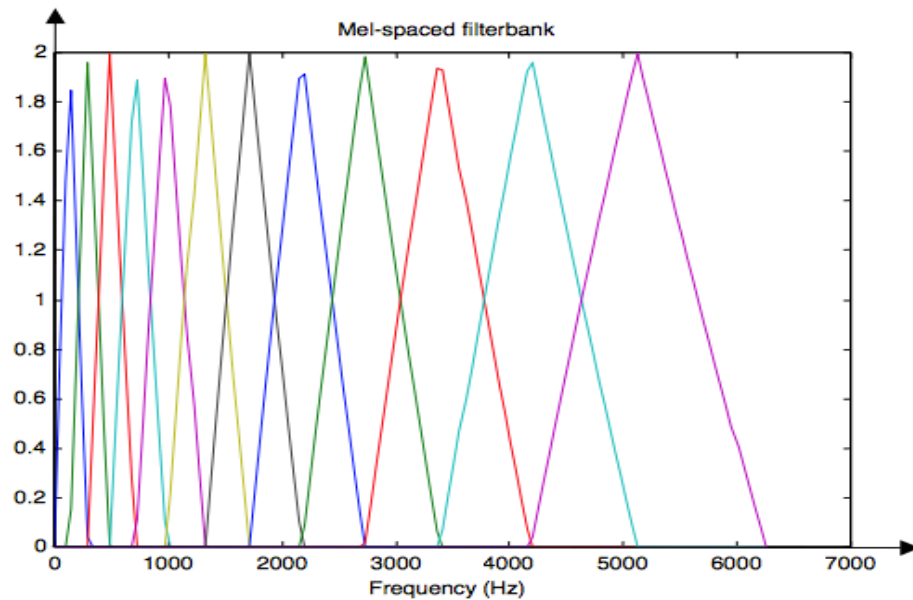


Figure 14: Mel-scale filter bank, (Tallat, Shahid, Samad, & Abbasi, 2014, p. 7).

Logarithm and discrete cosine transform

In this process, the log mel spectrum is converted into the time domain. This can be achieved by using a discrete cosine transform (DCT). The result of the conversion is called mel frequency cepstrum coefficient and the set of coefficients are acoustic vectors. Therefore, each input word is converted into a series of acoustic vectors (Borde, Varpe, Manze, & Yannawar, 2015).

2.2.2 Linear prediction cepstral coefficients

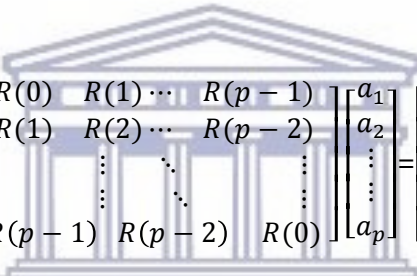
Linear prediction cepstral coefficients are some of the earliest speech feature extraction algorithms, see Figure 15. They are based on predicting the current sound sample as a linear mix of past acoustic samples, and assuming that the audio device's shape governs the nature of the sound produced (Bahattarjee, 2013). This method is an attempt to mimic human speech by working at a low bit-rate and was derived using an autocorrelation method (Veton Z & Hussien A., 2015).

Autocorrelation is a technique used to find the correlation between the signal and itself. The autocorrelation method evaluates the LPCC set and thus achieves filter efficiency (Dave, 2013). The set of linear prediction coefficients specifies the characteristics of the vocal tract transfer. A digital all-pole filter models the vocal tract, and the transfer function in the z-domain is given by:

$$V(Z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (6)$$

Where $V(Z)$ is the vocal tract transfer function, G is the gain filter and (a_k) is a set of autocorrelation coefficients.

Autocorrelation involves the calculation of a matrix of simultaneous equations and the autocorrelation of the windowed speech frames. The matrix of equations that need to be solved is:



$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(2) & \cdots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix} \quad (7)$$

where $R[n]$ is the autocorrelation function of a window speech signal.

The gain of the all-pole can be found by solving the following equation:

$$G = \sqrt{R[0] - \sum_{k=1}^p a_k R[k]} \quad (8)$$

The cepstral coefficient is calculated from LPC through recursive algorithms. The cepstral analysis operation is a process to find the cepstrum of a sound sequence, as given below. This method is called LPCC (Rabiner & Schafer, 1978)

$$C[n] = a_n + \sum_{k=1}^{n-1} \binom{k}{n} C[k]a_{n-1}, 0 \leq i \leq p. \quad (9)$$

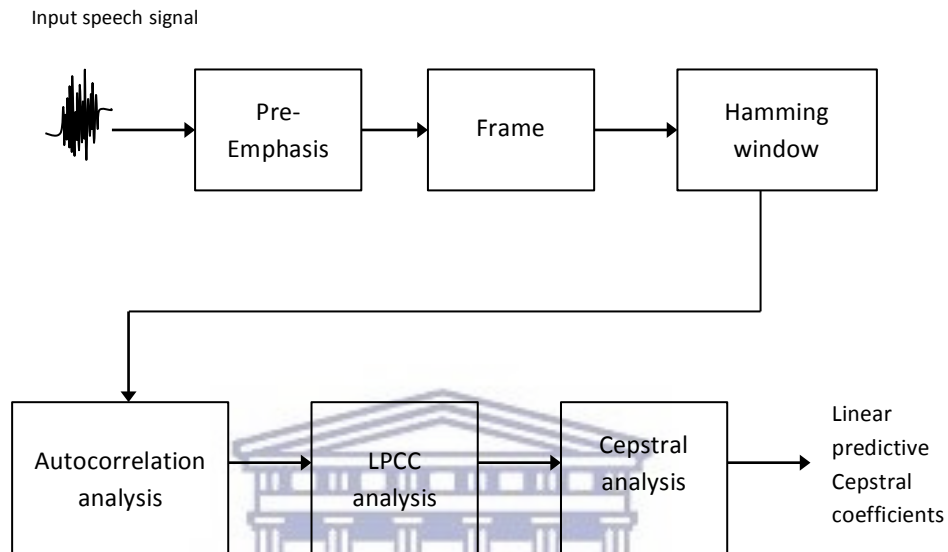


Figure 15: Linear prediction cepstral coefficients (LPCC) adapted from (Veton Z & Hussien A., 2015, p. 4).

2.2.3 Perceptual linear prediction

Perceptual linear prediction (PLP) is based on the power-law of nonlinearity. Harmansky proposed PLP in 1990 (Harmansky, 1990) see Figure 16. In the next stage, in order to obtain the linear prediction (LP) analysis, an Inverse Discrete Fourier Transform (IDFT), is used as a perceptual autocorrelation sequence (Alam, Kinnunen, Kenny, Ouellet, & O'Shaughnessy, 2013).

To obtain the final features from the LP coefficients, an auditory-like kestrel recursion is performed (Gold, Morgan, Ellis, & O'Shaughnessy, 2012).

In this type of extraction the features of the sound are achieved by performing spectral analysis, frequency band analysis, equal loudness pre-emphasis, intensity-loudness power law and autoregressive modelling. The hidden Markov model toolkit (HTK)-based processing is followed for PLP feature extraction, and for auditory frequency analysis. Instead of a trapezoidal-shaped bark filter bank, a mel filter bank is used (Alam, Kinnunen, Kenny, Ouellet, & O'Shaughnessy, 2013).

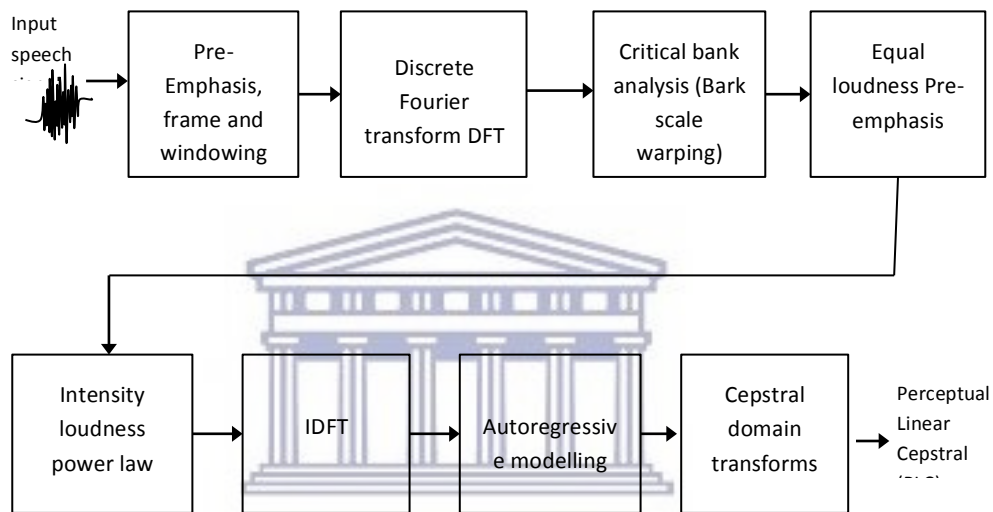


Figure 16: Perceptual Linear Prediction (PLP), Adapted from Venton & Hussien (Veton Z & Hussien A., 2015, p. 5).

2.3 Feature Matching (Classification)

Feature matching is an aspect of pattern recognition. In general, a pattern is the object of interest. A pattern sequence of an acoustic vector is extracted from speech input using one of the feature extraction algorithms; these patterns are used in the classification algorithm.

Feature matching is the most important aspect of speaker recognition. This procedure involves identifying an anonymous speaker by comparing the extracted features from the speech signal—obtained using one of the feature extraction algorithms—to a set of features of known speakers (Tallat, Shahid, Samad, & Abbasi, 2014).

Many methods have been proposed for speaker recognition modelling and matching, however the most popular methods for *text-independent speaker recognition* are:

- Support Vector Machines (SVM)
- Vector quantization (VQ)
- Gaussian mixture model (GMM)
- Artificial neural network (ANN) and
- Fully-ergodic hidden Markov model (HMM)

In addition to the previous methods, the following methods are common methods for *text-dependent speaker recognition*:

- Hidden Markov model (HMM)
- Dynamic time warping (DTW) (Salna & Kamaraukas, 2015) (Loh & Abdul Manan, 2010).

These methods were used for speaker recognition in many studies by different researchers for example Kaur who used GMM for feature classification. His system gave a high accuracy of 99% (Kaur & Kaur, 2016). On the other hand, according to Parul, who used GMM, HMM and VQ for feature classification in a speaker recognition system, the system tends to have accuracy between 79% to 87% (Parul, 2012).

In the next sections these methods will be discussed in more detail.

2.3.1 Support Vector Machines

SVMs were introduced by Boser, Guyon and Vapnik for classification problems. SVMs were derived from statistical learning theory in the 90s. It is a relatively new machine learning method. SVM is an effective feature classification model. It gave good performance empirically in numerous successful applications fields for example biometrics, text recognition, voice and image processing, etc. (Weston, 2014). Although a SVM is complex mathematically and arithmetically expensive, it is one of machine learning techniques, that help solving huge data classification problems in multi-domain applications (Sutharan, 2016). SVMs are derived from the theory of statistical learning and are applied as a

machine-learning tool that inherently classifies data into two classes. SVMs offer several advantages such as dealing with high-dimensional feature vectors without affecting the training time; being memory efficient by only using a subset of training points in the decision function; and using different kernel functions that offer both power and flexibility (Whitehill, 2006).

SVMs provide a classification learning model and an algorithm rather than a model. The algorithm allows linear domain division using a simple and then manipulates it (Hearst, Dumais, Osman, Platt, & Scholkopf, 1998). The kernel is the linear kernel that can easily be substituted with the radial basis sigmoid, polynomial or other more recent kernels that allow features to be clearly in a given classification problem. Alternative kernels allow SVMs to classification problems using linear classification techniques (Tzotsos, 2006), see



UNIVERSITY *of the*
WESTERN CAPE

Figure 17.

The SVM aims to maximize a mathematical function given a set of data points (Noble, 2006). Consider a set of data points that consists of two classes: the theory of SVMs suggests that it is possible to find a boundary that can separate the two classes. Furthermore, consider a training set of data points, N , that is represented by $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where each x_i , with $i = 1, 2, \dots, N$, is a data point in R^n and each $y_i \in \{-1, 1\}$ is the corresponding classification label for x_i such that the data points are divided into positive and negative classes. Moreover, suppose that the positive class, $S = \{x_i | y_i = 1\}$, and negative class, $S = \{x_i | y_i = -1\}$ are linearly separable in R^n , such that at least one boundary can be formed between the two; this boundary is referred to as the decision boundary (Noble, 2006).



UNIVERSITY of the
WESTERN CAPE

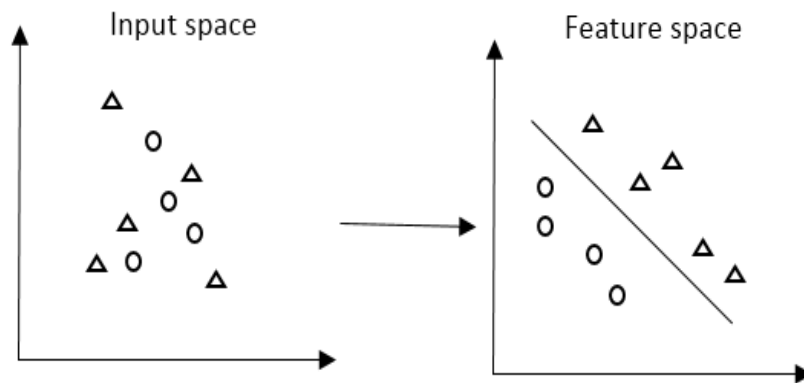


Figure 17: Support Vector Machine.

In a higher dimensional space the boundary is regarded as a geometrical concept of a plane, and is referred to as a hyperplane. See Figure 18, where the hyperplane is defined by the following equation:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0; \mathbf{w} \in R^n, \mathbf{b} \in R. \quad (10)$$

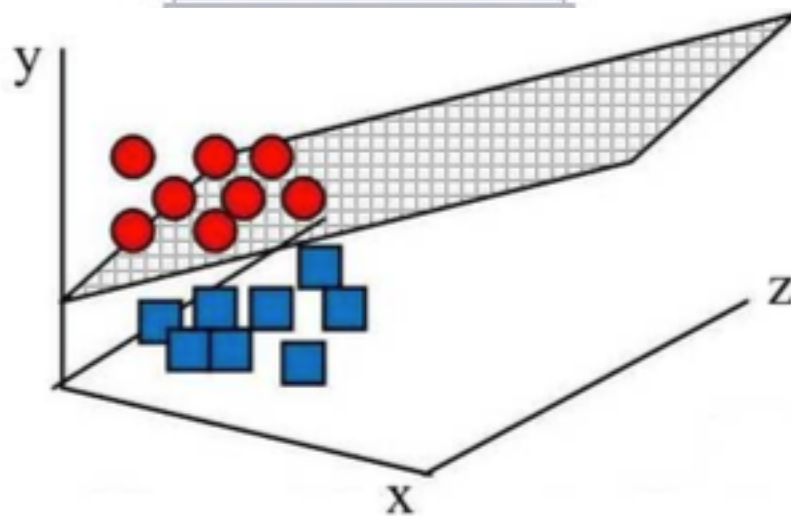


Figure 18: Linear classification of hyperplane. (Achmed, 2014, p. 57)

Where w is the normal vector, x is the feature set and b is the interim term. The normal vector, w , of the hyperplane is defined as a linear combination of data points, x_i , with weights, a_i , and expressed as:

$$\mathbf{w} = \sum_{1 \leq i \leq N} a_i x_i y_i. \quad (11)$$

The two hyperplanes can be selected if the data points are linearly separable. The hyperplanes can be described by the following equations:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} + \mathbf{b} &= +1, \\ \mathbf{w} \cdot \mathbf{x} + \mathbf{b} &= -1. \end{aligned} \quad (12)$$

Midway between these hyperplanes, there is a decision boundary that can be defined as:

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0. \quad (13)$$

In addition, the distance between the margin and the decision boundary can be described as:

$$d = \frac{2}{\|\mathbf{w}\|}. \quad (14)$$

The selection of a hyperplane can be determined by two factors:

- The data points should separate clearly.
- The hyperplane should have the same maximum distance to the nearest data point from both classes.

This distance is referred to as the *margin* and the data points that are situated closest to the hyperplane are referred to as the support vectors. It is necessary to find the maximum margin if the separation between the two classes is the greatest, because this will allow the SVM to classify new data points more accurately. To determine this hyperplane, two requirements need to be met:

1. All training data points should be classified correctly, where w and b should be estimated such that:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + \mathbf{b}) \leq -1, \text{ for } y_i = -1, \quad (15)$$

and

$$y_i(\mathbf{w} \cdot x_i + \mathbf{b}) \leq 1, \text{ for } y_i = +1. \quad (16)$$

Combining these two equations gives:

$$y_i(\mathbf{w} \cdot x_i + \mathbf{b}) - 1 \geq 0, \forall i = 0,1,2, \dots, N. \quad (17)$$

2. The margins should be as large as possible; maximising the distance in equation 13 is the same as minimising $\frac{\|\mathbf{w}\|}{2}$. This results in minimising:

$$f(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2.$$

The optimum hyperplane can therefore be found by solving the optimisation problem (Long, 2006), which is defined as:

Minimise

$$\frac{1}{2} \|\mathbf{w}\|^2, \quad (18)$$

subject to

$$y_i(\mathbf{w} \cdot x_i + \mathbf{b}) - 1 \geq 0, \forall i = 0,1,2, \dots, N. \quad (19)$$

The optimisation problem can be translated to:

Maximise

$$\sum_{i=1}^N a_i - \frac{1}{2} \sum_{i,j=1}^N a_i a_j y_i y_j (x_i, x_j), \quad (20)$$

subject to:

$$\sum_{i=1}^N a_i y_i = 0, \quad (21)$$

where $a_i \geq 0$ and $i = 0,1,2, \dots, N$.

The hyperplane selected is referred to as the maximum margin; the optimal hyperplane discriminant function can be defined as:

$$f(x) = \sum_{i \in S} a_i y_i (x_i x) + b. \quad (22)$$

where S the subset of support vectors that corresponds to positive Lagrange multipliers.

The linear classification divides the data domain linearly, i.e., by a straight line or hyperplane, to separate the classes in the original domain. In LSVM the data domain is

divided and mapped into a response set (Hastie, Tibshirani, & Friedman, 2009). The modelling of a LSVM adopts the following linear equation:

$$Y = WX' + \gamma. \quad (23)$$

In nonlinear classification the data domain can be transformed to a feature space that can be divided linearly. The nonlinear SVM includes mapping of the data domain to feature space using a kernel function (Scholkopf, et al., 1999), mapping the feature space domain into a response set, then dividing the data domain (Hastie, Tibshirani, & Friedman, 2009), The modelling of nonlinear SVM adopts the following linear equation:

$$Y = W\phi(X') + \gamma. \quad (24)$$

Using a Support Vector Machine for classification includes parameterization and the optimization objectives. These objectives are based on the topographical structure of the class in the data domain. This means the classes may be linearly detachable or linearly non-detachable. Therefore, the parameterization and optimization objectives must consider these class properties carefully (Huang, Chen, Zhou, & YinandK.Guo, 2011). See Figure 19.

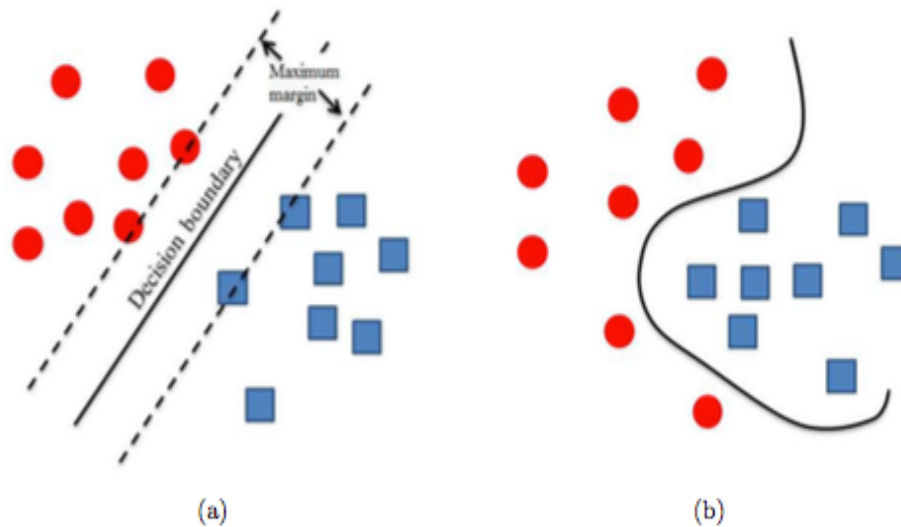


Figure 19: (a) Linear classification, (b) Non-Linear classification. (Achmed, 2014, p. 56)

The scalar product of the vectors can be formed in the mapping space in the same way as the function of the scalar products of the corresponding vector in the current space (Tzotsos, 2006). This equivalence can be expressed as:

$$\begin{aligned}
 K(x_i, x_j) &= \Phi(x_i) \cdot \Phi(x_j), \\
 &= (x_i, x_i^2) \cdot (x_j, x_j^2), \\
 &= x_i x_j + x_i^2 x_j^2 \\
 &= x_i \cdot x_j + (x_i, x_j)^2.
 \end{aligned} \tag{25}$$

where $K(x_i, x_j)$ represents the kernel function.

Any set of data points can be separated linearly into a higher-dimensional space without knowing the explicit form of Φ , by selecting the appropriate kernel.

Thus the problem of dual optimization can be formulated as:

Maximise:

$$\sum_{i=1}^N a_i - \frac{1}{2} \sum_{i,j=1}^N a_i a_j y_i y_j K(x_i, x_j), \tag{26}$$

subject to:

$$\sum_{i=1}^N a_i y_i = 0, a_i \geq 0, \tag{27}$$

where $i = 1, 2, \dots, N$.

Thus, the decision function becomes:

$$f(\mathbf{x}) = \sum_{i \in S} a_i y_i K(x_i \cdot \mathbf{x}) + \mathbf{b}. \tag{28}$$

where K is the kernel function, \mathbf{b} is the interim term and S is a set of support vectors.

Using one of the kernel functions that map data from a current space onto higher-dimensional feature space, can give an optimal hyperplane that separates the classes. There are four common kernel functions that are used for nonlinear feature mapping (Moreno & Ho, 2003):

1. Radial basis function (RBF) Gaussian: $K(x_i, x_j) = \exp(-\gamma \cdot \|x_i, x_j\|^2)$. $\gamma > 0$.

2. Polynomial function: $K(x_i, x_j) = (x_i^T x_j + r)^d$, where d is the degree of the polynomial.
3. Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.
4. Linear: $K(x_i, x_j) = x_i^T x_j$,

where γ , r and d are kernel parameters.

2.3.2 Vector quantization

Vector quantization (VQ) is a powerful method for multimedia communications and is based on the principle of block coding. It was proposed in 1980 by Linde, Buzo and Gray (LBG). In the VQ process a large number of feature vectors are taken to produce a smaller set of measured vectors that represents the centroids of its distribution (Tiwari, 2010). The VQ design algorithm is based on training quantization (Nijhawan & Soni, 2014), (Ku, Chang, & Hwang, 2014).

Srinivasan has defined VQ as the process of dividing a large vector space into a limited number of regions; where each region is called a cluster. All clusters are their centre; this centre is called a code word. All code words are codebook, combined with a background model to provide competitive often used for computational speed-up techniques (Srinivasan, 2012). See

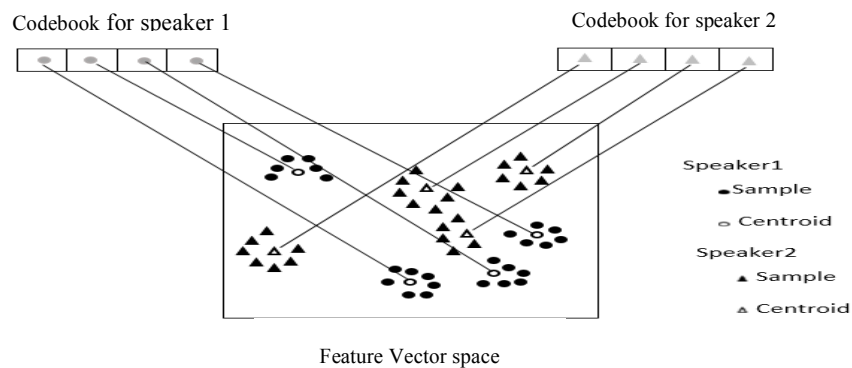


Figure 20.

Figure 20: Vector quantization codebook formation. Adapted from (Tallat, Shahid, Samad, & Abbasi, 2014, p. 9)

2.3.3 Gaussian mixture model

The Gaussian mixture model (GMM) is a probabilistic model. It is assumed that all data points are derived from a mixture of a finite number of Gaussian distributions with unknown parameters. Mathematically GMM is a density model, which is represented by M Gaussian component densities. The functions of these components are combined to provide multimodal density. Parametric models of some arbitrary probability distributions are often utilized in GMMs. The GMM parameters can be obtained by the expectation-maximization (EM) algorithm to train the data (Silveira, Schroeder, da Costa, de Oliveira, Junior, & Junior, 2013). GMMs are widely used in recognition systems such as biometrics recognition, which are related to vocal-tract and spectral features analysis, and in emotion recognition systems; GMMs use probability distribution features (Utane & Nalbawar, 2013). The probability distribution can be expressed by the following equation (Nakagawa, Asakawa, & Wang, 2007):

$$p(x/\lambda) = \sum_{i=1}^M p_i b_i(x), \quad (29)$$

where M denotes the number of mixture weights, x is the continuous-valued data vector, p_i is the component weight and $b_i(x)$ is the component density (Kamarauskas, 2015).

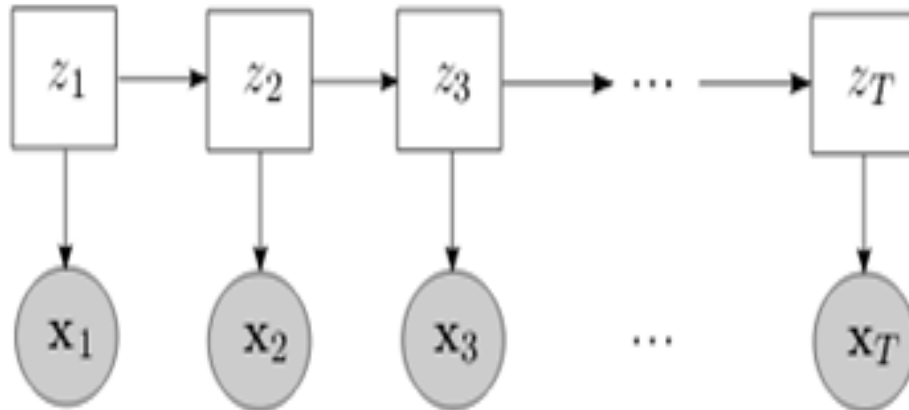
2.3.4 Artificial neural network

A neural network is a computing system consisting of a number of simple, interrelated processing elements, that respond to an external input (Caudill, 1989).

An artificial neural network (ANN) solves problems by learning based on combining artificial neurons to do information processing. The weights of artificial neurons are adjusted to get a particular output from a particular input, where adjusting the weights during processing is known as learning (Shah, 2009).

2.3.5 Hidden Markov model

The hidden Markov model (HMM) is a statistical approach commonly used. HMM was developed by Baum and his colleagues in the late 1960s and early 1970s. It can be used in



many fields such as bioinformatics, econometric studies and population genetics (Yildirim, Singh, Dean, & Jasra, 2015). It has been used for modelling observed patterns from the 1970s, and has been applied on a large scale in speech processing applications since the mid-1980s (Zhang, Sun, & Luo, 2014).

Figure 21: Hidden Markov model.

The HMM consists of a Markov chain defined by Bonneville and Jin as a type of stochastic finite-state machine of detached hidden variables and a series of constant observed variables, each of which is dependent on one detached variable's state (Bonneville & Jin, 2013) (Ghahramani, 2001).

The structure of the HMM is shown in Figure 21 where the sequence Z_1, \dots, Z_T represents the detached hidden variables. The value of each detached variable Z_T corresponds to one of K states, while the sequence x_1, \dots, x_T represents the continuous observed variables. The value of each one of the continuous observed variables can take on any real number (Zilli, Parson, Merrett, & Rogers, 2014).

2.3.6 Dynamic time warping

Dynamic time warping (DTW) can be computed in quadratic time using dynamic programming. DTW is a useful solution for time-series problems in a variety of domains (Tarango, Keogh, & Brisk, 2014). It can be applied in many applications such as speech recognition systems, bioinformatics, data mining and hand writing recognition. It was introduced for speech processing in 1968. DTW is an algorithm to determine the similarity of time discrete signals or feature sequences sampled at equidistant points in time. DTW aims to achieve maximal similarity between two sequences of feature vectors by warping the time axis repeatedly to achieve the optimum fit (Celebi, Aydin, Temiz, & Arici, 2013) (Grutzmacher, Wolff, & Haubelt, 2015). Figure 22 shows how DTW aligns two time series by warping the time dimension.

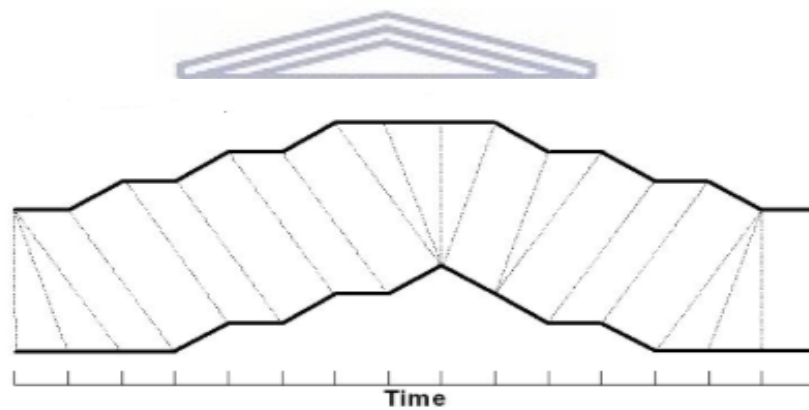


Figure 22: Dynamic Time Warping (DTW). (Muda, Begam, & Elamvazuthi, 2010, p. 140).

2.4 Summary

In the preceding sections an overview of speaker recognition algorithms was given, and each was described and explained. An overview of several approaches was given and the aspects of these that researchers consider useful when analysing speaker identification, were explained.

In the following chapter the approach and methodology used for this research will be described.

Chapter 3 Research Methodology

In the previous chapter, concepts related to speaker identification were explained, the contribution of researchers in the domain of voice processing was investigated and some of the common speaker recognition techniques were reviewed. In this chapter, the philosophical grounding that underpins the research is discussed to ensure the consistency of the study, the methodology, which is informed by a philosophical stance and the methods, which are selected to perform the data analysis, are discussed.

3.1 Research Philosophy

Research philosophy has been defined by Collis and Hussey as a belief about the way in which data about a phenomenon should be gathered, analysed and used. At the same time all research is based on latent assumptions about what constitutes valid research, whether it is quantitative, qualitative or both, and which research methodologies and methods are appropriate. It should be noted that some researchers use the terms 'methods' and 'methodology' interchangeably (Collis & Hussey, 2013). In this research methodology refers to the overall approach or design that lies behind the selection of specific methods, and that links the selection of methods to the desired end result.

To ensure the consistency of the study, it is necessary to consider the philosophical basis that informs the decision-making process. According to Crotty (1998) four basic building blocks frame the research process: epistemology, theoretical perspective, methodology and methods (Crotty, 1998). See Figure 23.

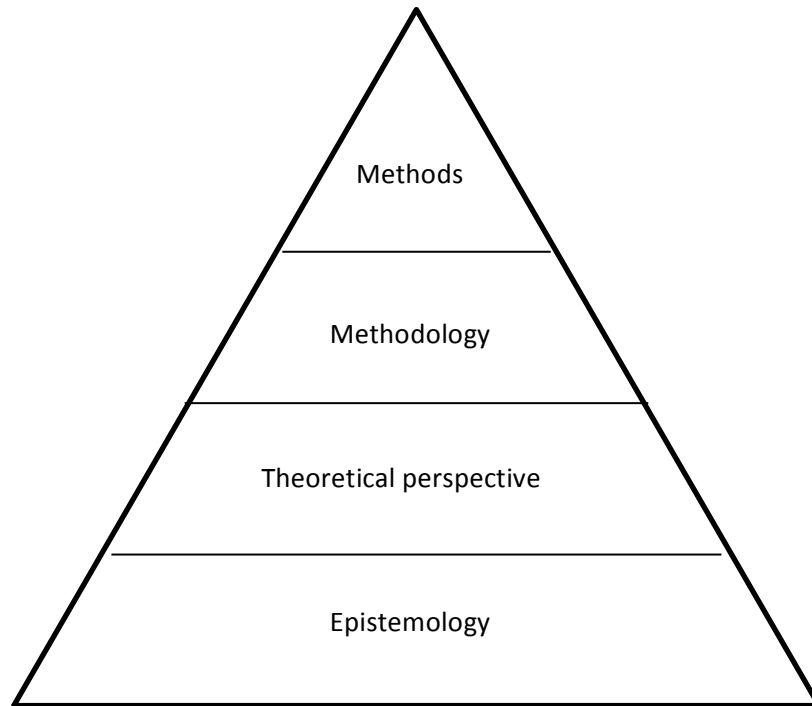


Figure 23: The four basic elements of research according to Crotty (1998).

The decision-making process in the study can be defined by posing four questions that relate to the four basic elements:

1. What methods are to be used?
2. What methodology guides the choice and use of the proposed methods?
3. What theoretical perspective underpins the preferred methodology, and
4. What epistemology informs the suggested theoretical perspective?

Postulating the research process in terms of these four elements ensures the soundness of the research and maintains consistency within the study. It also forms an analysis of the process, points out the theoretical assumptions that underpin it and determines the quality of its findings. The four basic elements of Crotty's paradigm will be discussed in more detail in Sections 3.1.1—3.1.4.

3.1.1 Epistemology

Epistemology refers to theory of knowledge that defines what kind of knowledge is possible and legitimate. It is combined with ontology, which is defined as the study of being and deals with ways of constructing reality and describes our views, claims or assumptions on the nature of reality (Flowers, 2009). It provides a philosophical grounding for the decision regarding the kind of knowledge that is possible and ensures that it is both adequate and legitimate, although a range of epistemologies exist (Crotty, 1998).

Each epistemology contains assumptions about the nature of the world and these assumptions are in turn embedded in the particular methods. Crotty suggests three epistemological stances: objectivism, subjectivism, constructivism and their variants (Feast, 2010).

Objectivism maintains that knowledge exists whether we are conscious of it or not. The existence of knowledge is independent of our consciousness of it. With objectivism, researchers look for explanations by developing and testing hypotheses and theories (Feast, 2010) (Biggs & Buchler, 2008).

Constructionism maintains that meaning is constructed through our minds interacting with the world, which implies that people in different cultures or eras construct meaning in different ways even in relation to the same phenomenon (Crotty, 1998).

Subjectivism, the last stance, maintains that knowledge is generated from the mind, without reference to reality. While constructivism acknowledges the influence of (the) reality in the generation of meaning, subjectivism holds that gaining knowledge about the world is done through introspection (Feast, 2010).

The current study is done from an objectivist epistemological stance.

3.1.2 Theoretical perspective:

A theoretical perspective is described as a research paradigm or research philosophy and refers to the philosophical assumptions related to the underlying epistemology that guides the research (Flood, 2010).

Two research paradigms have been suggested by Rubin & Rubin, these paradigms are: positivism and naturalism, which differ in terms of the goals of the research and the ways that are used to achieve the goals. Positivists claim the existence of a single and objective reality or truth that can be observed and directly measured, the purpose of a positivist is to work out theories and prove or disprove their hypotheses (Rubin & Rubin, 2012).

Naturalists argue that reality is indirectly measurable, i.e., reality can be changed and perceived differently through the interpretations of people. Naturalists describe and explain a complex situation or process. Their purpose is more to explain and understand what has happened in a specific circumstance, than to prove or disprove a hypothesis. To differentiate between the two stances Rubin & Rubin (2012) have proposed a list of questions:

1. Is the purpose to test theories and discover general principles, or is it to describe and explain complex situations?
2. Should the work be primarily deductive; that is, should it start out with broad theories and suppositions and then systematically test their implications?
3. Or should it be inductive, i.e., should it build explanations from the ground up, based on what is discovered?
4. Is there one truth out there that the researcher is trying to measure, or are there many possibly contradictory ones?

The research considerations with regard to the current study followed a positivist theoretical perspective, since the study has a more objective approach. This particular paradigm offers a range of methodological choices where researchers can apply quantitative methods.

Methodology

A research methodology is the system used to address a research problem. It is a set of procedures to solve or describe specific phenomena concerned with the problem (Rajasekar, Philomonathan, & Chinnathambi, 2006). Thus a methodology refers to a

strategy of investigation or a work plan that moves from the underlying philosophical assumptions to the research design and data collection.

For this research the design science research methodology of Brocke & Buddendick (2006) is employed.

Design science research

Design science research (DSR) is a research paradigm used when artefacts are developed to solve a problem. The knowledge derived during the process, is recorded and contributes to the field of knowledge. DSR is an iterative process where each cycle consists of six phases: identify, build, document, select, evaluate and communicate (Vom Brocke & Buddendick, 2006) as shown in Figure 24.

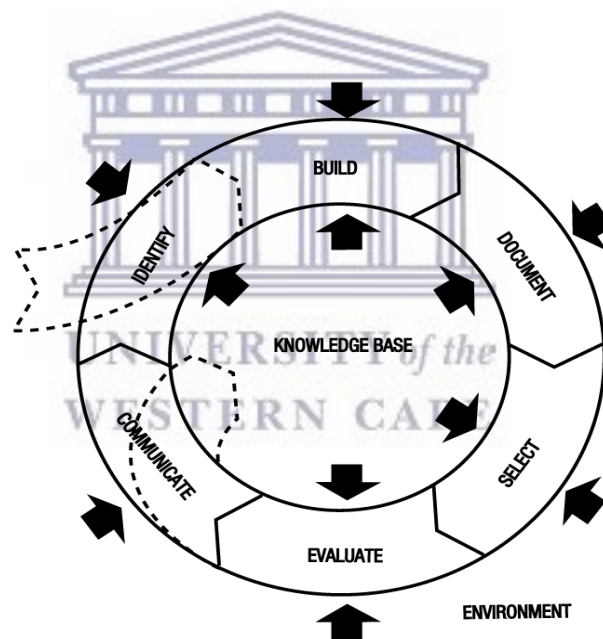


Figure 24: The iterative process of a general DSR cycle consists of six basic stages (Vom Brocke & Buddendick, 2006, p. 582)

The key aspects that need to be considered in each phase can be summarized as follows:

- **Identify:** this phase should answer the following questions:
 - What is the problem?
 - How should the problem be solved?
 - What are the specific criteria that a solution for the problem should meet?

The researcher needs to identify the research problem clearly and needs to justify the value of a solution. In addition, the objectives of the research and the derivation of the requirements are described in this phase. Perhaps the main focus in this phase is to understand the problem's relevance, together with current solutions and their weaknesses.

- **Build and document:** in this phase an artefact, capable of solving and delivering the functionalities is developed. Building the artefact can be achieved using constructs, models, methods, or instantiations. The research contribution is embedded in the application of methods, technologies and theories to create the artefact that delivers utility.
- **Select and evaluate:** the principal aim of this phase is to determine how well the artefact works. After the design of a solution, the use of the artefact to solve one or more instances of the problem should be illustrated. Before the evaluation, the evaluation criteria and techniques should be chosen, then the system is tested and analysed according to the selected evaluation metrics.
- **Communicate:** this is a comparison phase, where the problem solution, its novelty and effectiveness are compared to other work and communicated to relevant audiences.

The results obtained after each cycle can serve as new requirements to solve the next instance of the problem.

3.1.3 Methods

Research methods refer to the techniques or procedures that are used to collect and analyse data related to the research question. In this study three different techniques are used to achieve the aim of the research:

- a visual representation method which uses the voice image to recognize individuals.
- a feature extraction and matching method that uses the MFCC algorithm for extractions and the SVM algorithm for feature classification—MFCC+SVM.
- a feature extraction and matching method that uses the LPCC algorithm for extraction and the SVM algorithm for feature classification—LPCC+SVM.

3.2 Research Design

3.2.1 Application of design science research

In order to answer the research question, the DSR cycle was applied as three cycles or phases of analysis: the visual representation-phase, the MFCC+SVM phase and the LPCC+SVM phase. See Figure 25.

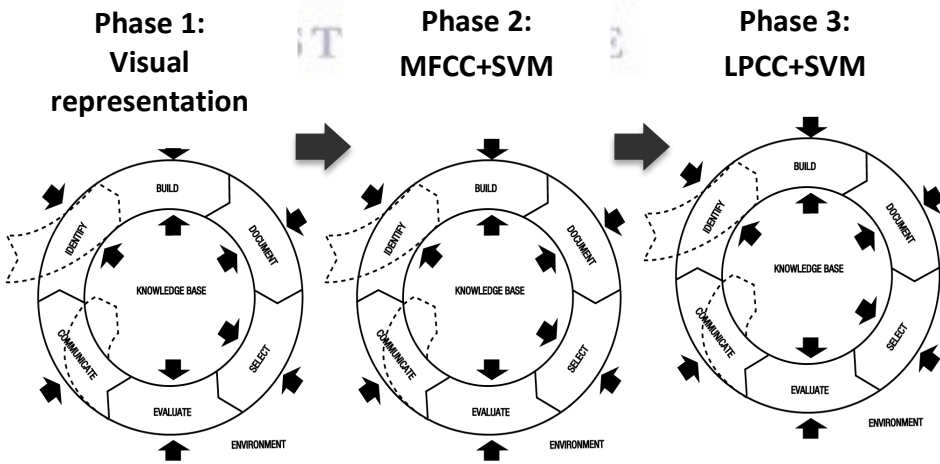


Figure 25: The DSR cycle applied to each of the phases.

Phase 1: Visual representation cycle/phase

Data collection process

For the first cycle of the research, the UWC Research Committee gave ethical clearance for data collection as well as the experiments with randomly selected participants (reference number 15/7/242). Participants, students at the University of the Western Cape, were advised of the purpose of the study, their right to withdraw and their right to anonymity. Those that agreed to be part of the study, were asked to read a specific paragraph, which was recorded, and the voice recordings were used as input to the application designed for the research. As explained to the participants, their recorded voice messages would not be linked to them in any way and would be destroyed once the research was completed. No vulnerable persons were involved in the research.

Audio data was collected from six adults, 3 females and 3 males, aged between 26 and 50, in a quiet environment using a normal microphone and a MacBook Pro. All speakers repeated the same phrase four times to create 24 voice images. These voice images were used for visual analysis experiments. During this cycle the visual representation of the voice signature, thus a voice image, was considered.

Method used

The voice signatures were transformed into voice images using a program called Audacity⁵, Which is a computer application used for digital audio editing and recording. It can be used for post-processing of all types of audio file formats such as .wav, .MP3 and podcasts, by adding effects such as normalization, trimming and fading in and out. In this research, Audacity was used to normalize the waveform of each voice image and also to produce a linear frequency analysis image of each voice image. See Figure 26. The linear frequency analysis image was preferred for this experiment since it does not need any normalization. The Audacity software produces the same image size for each voice image, although this is not the case for the waveform of each voice image. The waveform sizes differ, because the talking speed is different for each voice image.

⁵ <https://multimedia.journalism.berkeley.edu/tutorials/audacity/>

Linear frequency analysis images of the 24 voice images were created using Audacity and saved in a database. This database was created using a structured query language (SQL) database. VB.net was used to create a program to compare the images. The program comparing the voice's images first checks the size of the two images, then compares the images pixel-by-pixel to get the percentage of similarity between pairs of images.

The similarity was measured using the Jaccard similarity index $J(M, N)$ as following:

$$J(M, N) = \frac{|M \cap N|}{|M \cup N|} \times 100$$

$$= \frac{|M \cap N|}{|M| + |N| - |M \cap N|} \times 100$$

where the index is a 100 when M is exactly equal N, and 0 when the intersection is empty⁶. Depending on this percentage, the program decides whether or not these two images belong to the same speaker.

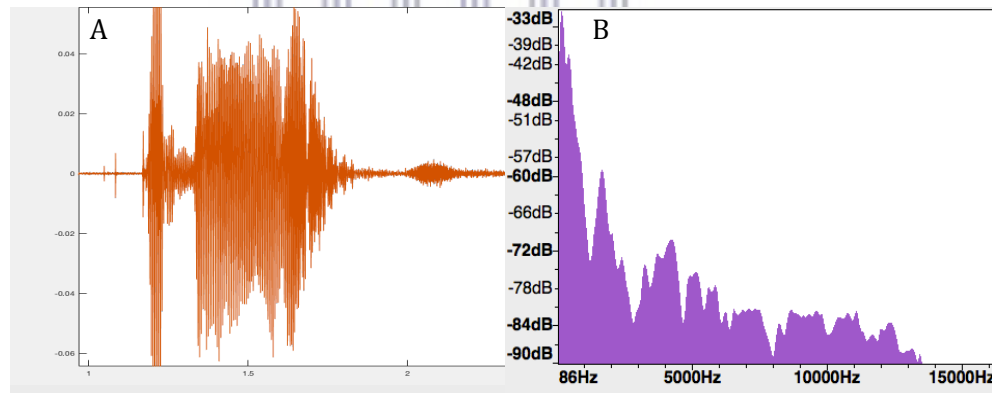


Figure 26: A. Voice waveform, B. Linear frequency analysis image.

Experiments were carried out with the visual representation of the voice signatures. The images were compared to determine if the images are from the same speaker or not.

⁶ www.statisticshowto.com/jaccard-index/

Experiments two and three were done for speakers of different gender to see if the comparison differs depending on the gender.

This simple method was proposed for the initial cycle to check if it is possible to recognise speakers without using more sophisticated methods. More sophisticated methods were applied in the second and third cycles.

Phase 2: Mel frequency cepstrum coefficients combined with Support Vector Machine cycle/phase

Data collection process

The Texas Instruments (TI) and Massachusetts Institute of Technology (MIT) audio database, called TIMIT⁷, was used for training and testing. The TIMIT audio database contains 6300 voice recordings. These were recorded using 630 speakers speaking in the eight major dialects of American English. Each speaker repeated the same phrase ten times. This database was used for the testing as well as for the training of the system in this cycle.

Methods used

MFCC—a feature extraction algorithm—was used to extract a set of features from a voice signal. These features were used by the Support Vector Machine for classification.

The experiments using MFCC with SVM were implemented with MatLab. The MatLab signal and image-processing package was used. The signal processing and auditory toolbox were also required to access all the necessary MatLab voice processing functions.

First, a voice signal, using the steps suggested in Section 2.2.1—pre-emphasis, framing, Hamming window, fast Fourier transform, mel-scale filter bank and logarithm and discrete cosine transform—was read and turned into a sequence of features. Many of these tasks are provided by MatLab functions such as *wavread* to read a voice signal, *hamming* for Hamming windowing, *fft* for fast Fourier transform, *dct* for discrete cosine transform and *melfb* for mel-scale filter bank.

⁷ <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1>

Then the voice signal was divided into frames with an overlap length of 25 ms, the result was a matrix where each column was a frame of N samples from the original signal. Next each voice signal was transformed to the frequency domain by applying windowing and FFT. The result of this task is the spectrum. The last processing step was to convert the power spectrum into the mel frequency cepstrum using the *melfb* function.

Finally, all the pieces were put together into a single Matlab function *mfcc*, which performs the MFCC processing. This function extracted 13 dimensional feature vectors for each frame.

SVM for classification, using the Matlab machine learning and statistics toolbox, was then used.

First the SVM classifier was trained: the *fitcsvm* function from the statistics toolbox was used for this training and optional cross validation. The output of this function was the trained model consisting of the optimal parameters of the SVM algorithm, which were applied to classify the new data.

Then the *predict* function of the MatLab statistics toolbox was used to test new unseen data from the TIMIT dataset. The MFCC features and polynomial SVM kernel function were used to get satisfactory predictions. The SVM was able to find a hyperplane, which separated positive and negative samples.

Before testing the system with the new data, as any system involved with machine learning methods it was necessary to training system with a number of data. The system was trained with 65 speakers from the TIMIT database. In order to measure the average accuracy of using MFCC combined with SVM to recognize a speaker, 35 different speakers from the TIMIT database were selected and used for testing the system. The recognition of each speaker was attempted eight times. The results were assessed based on the number of false acceptances and false rejections giving the accuracy:

$$Accuracy = \frac{\sum True\ acceptances + \sum True\ rejections}{\sum Number\ of\ recognition\ attempts}$$

Phase 3: Linear prediction cepstral coefficients combined with Support Vector Machine cycle/phase

Data collection process

The same dataset TIMIT was used as in the previous cycle—Phase 2.

Methods used

LPCC feature extraction algorithms, commonly used for speaker recognition, were considered for doing the experiments in this phase, SVM was again used for feature classification.

This experiment was also implemented using MatLab. The TIMIT audio database was used for training and testing during this phase. The system was trained with 65 speakers from the TIMIT database and tested with 35 speakers, as was done in Phase 2.

The *lpcc* MatLab function from the auditory toolbox was used to extract the features in this cycle/phase for classification.

To measure the accuracy of the LPCC algorithm combined with the SVM algorithm, 35 speakers were selected from the TIMIT audio database for testing[, as previously stated]. The recognition of each speaker was done eight times. The results can be assessed based on the number of false acceptances and false rejections.

$$Accuracy = \frac{\sum True\ acceptances + \sum True\ rejections}{\sum Number\ of\ recognition\ attempts}$$

3.3 Summary

In this chapter, the philosophical grounding that underpins the research was described in order to ensure the consistency of the study. It described how the epistemology and theoretical perspectives inform the research process. It was concluded that in order to follow a more objective approach, a positivist theoretical perspective should be taken. The discussion concluded that the DSR methodology is relevant to managing this philosophical stance. In addition, the six-stage cycle of the DSR framework that structured the research was discussed.

The specific methods that were used to collect the data and run the experiments were explained and discussed. Two different datasets were considered in this research:

- a created database with a small number of records that was used for the visual representation method, and
- the TIMIT audio database, which contains 6300 records. This dataset was used for training and testing since for this part of the experiment, a big dataset is required.

The next chapter describes and summarizes the main results obtained for the experiments that were conducted.



UNIVERSITY *of the*
WESTERN CAPE

Chapter 4 Results

The previous chapter outlined the approach and methodology adopted for the research, and considered the four important elements that influence the way in which research is undertaken: epistemology, theoretical perspectives, methodology and methods. A design science research methodology was applied to address the research problem in an incremental manner. In this chapter the results for each method are discussed and compared.

The aim of this study is to recognize a person by his/her voiceprint. To achieve the goal of this study a set of methods were considered: (1) visual representation—which compares the user's voice image to images stored in a database, to determine if the image matches any of the voice images in the database; (2) an MFCC feature extraction algorithm combined with SVM for feature classification; and (3) finally an LPCC feature extraction algorithm combined with SVM for feature classification.

4.1 Phase 1: Jaccard Similarity Method

The first cycle of the research is shown in Figure 27. The visual representation of a speaker's voice image, using graphs of the linear frequency, was compared to a database of 24 voice images.

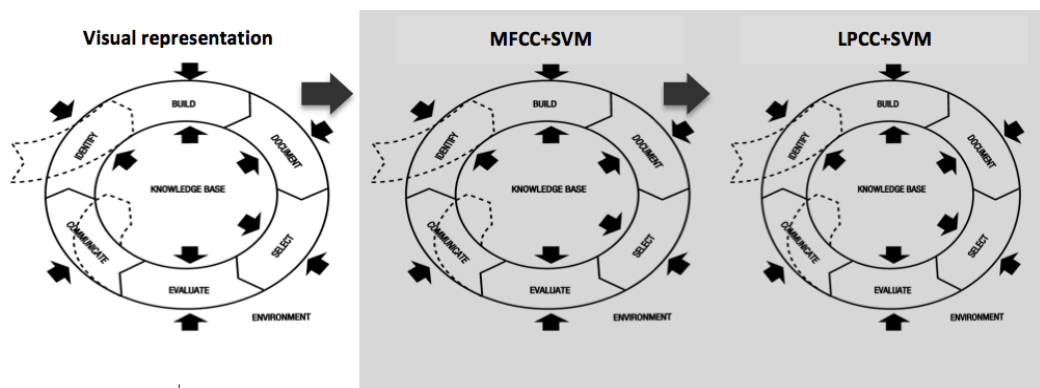


Figure 27: Phase 1: The first DSR cycle consisting of the visual representation phase.

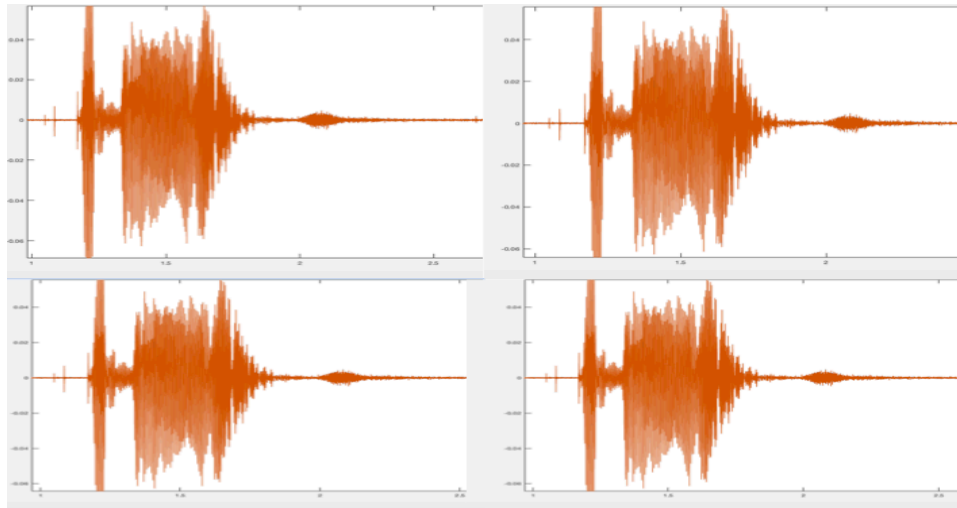


Figure 28: The four waveform images for Speaker 1.

Figure 28 depicts four voice image samples of the same speaker. As can be seen the lengths of time the speaker used to say the same phrase, were not the same.

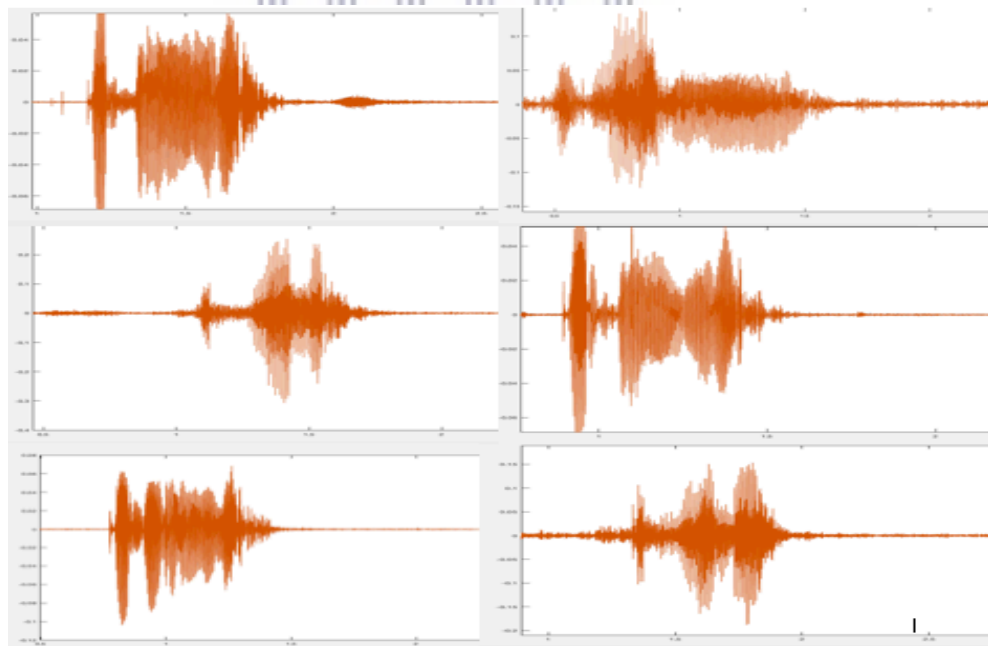


Figure 29: The waveforms of six speakers for the same phrase.

Figure 29 is a depiction of different speakers' waveforms and shows how the waveforms differ from speaker to speaker. To be able to compare the voice images, linear frequency images of the 24 images were created, using the Audacity software. The results were subsequently stored in a SQL database.

A linear frequency analysis image of Speaker 1 is shown in Figure 30.

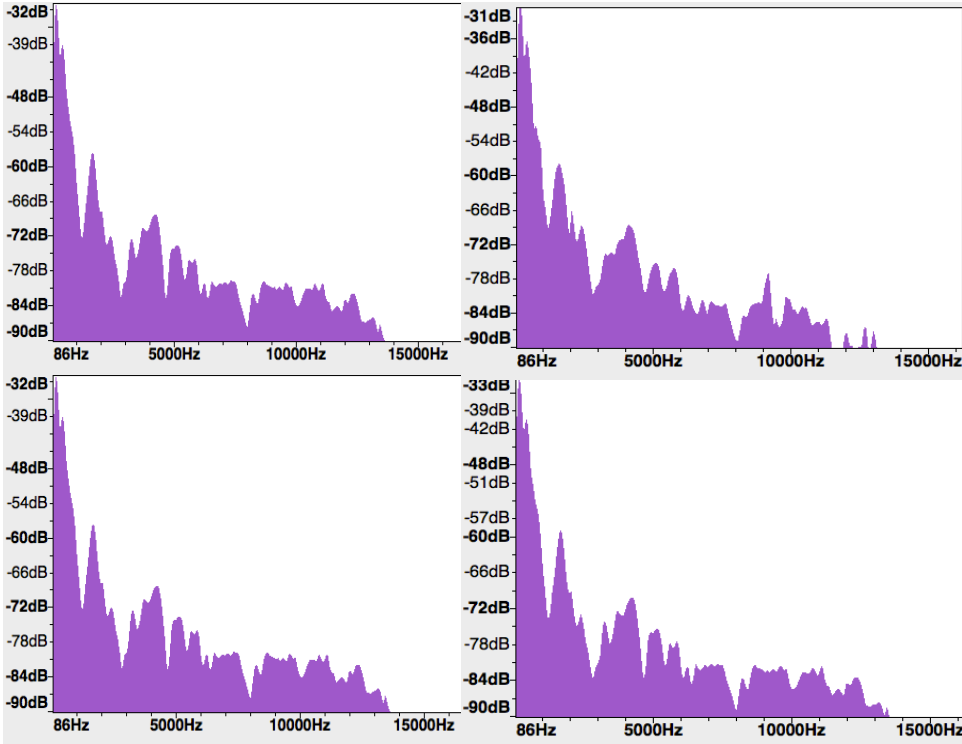


Figure 30: The four linear frequency analysis images of Speaker 1.

Figure 31 shows an example of one of the linear frequency analysis images (or waveforms) of each of the six person's voiceprints.

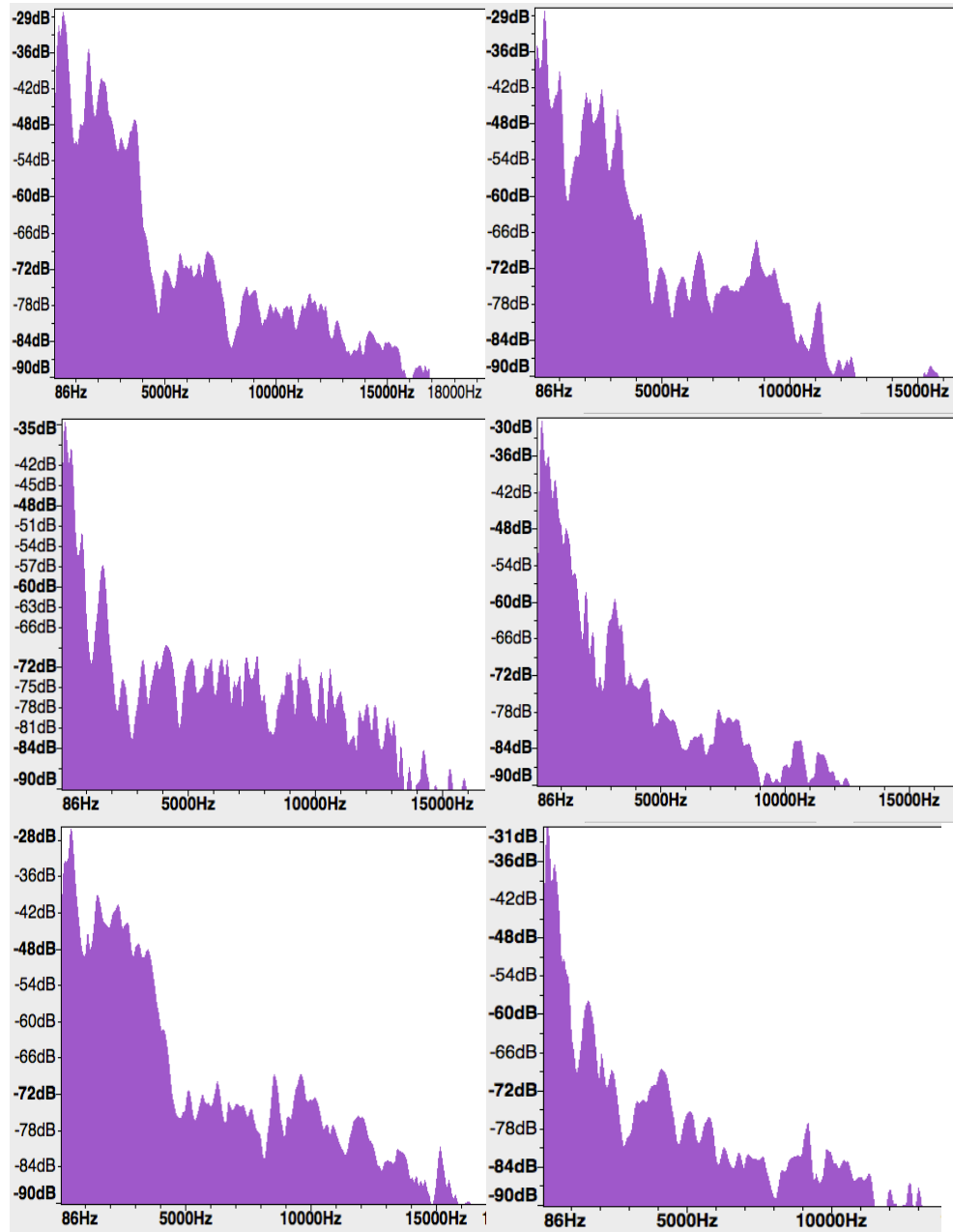


Figure 31: Linear frequency analysis images for six different speakers.

The linear frequency analysis voice images were compared as follows—the speaker with himself and the speaker with each of the other speakers—using a VB.net program to get

the percentage of similarity, which was measured by the Jaccard index, see Page 46. This experiment was executed to see if the program would be able to determine the identity of the speaker. The VB program compared the images pixel by pixel and then decided if the images belong to the same speaker or not. The results depend on the percentage of similarity between the voice images.

Three different experiments were done during this phase:

1. The comparison of the four voiceprints of the same speaker.
2. The comparison of Speaker 1's voice image—a female speaker—with the other five speakers' voice images.
3. The comparison of Speaker 3's voice image—a male speaker—with the other five speakers' voice images.

Experiment 1

For this experiment the following was done:

- The first image of Speaker 1 was compared with the second image, third image and fourth image of Speaker 1—thus three comparisons were done.
- The second image of Speaker 1 was compared to the third and the fourth image of Speaker 1—thus two comparisons were done.
- The third image of Speaker 1 was compared to the fourth image of Speaker 1.

Six comparisons were done for each speaker. Figure 32 depicts the six comparisons for each speaker for all six speakers. It shows the percentage of similarity between each two images for the same speaker, img1 VS img2 means the first voice image of the speaker compared to the second voice image, img1 VS img3 the first voice image of the speaker compared to the third voice image, etc. Each column depicts the percentage of similarity between the compared images. Each group belonged to one speaker.

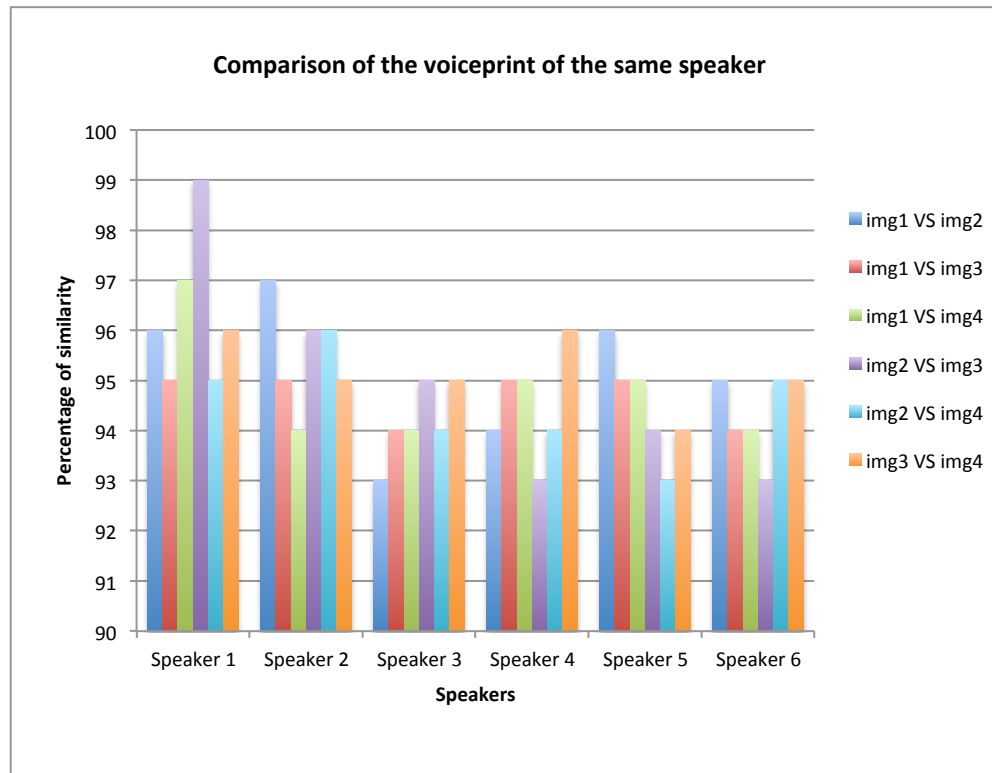


Figure 32: Six comparisons of the same speakers for all six speakers.

For example, the first group of data showed the similarity of the voice images of Speaker 1, which ranges between 95% and 99%.

According to data in the Figure 32 each speaker's voice image are extremely similar as one would expect, since the percentages range between 93% and 99%.

Experiment 2

In this experiment the four voice signatures of Speaker 1, a female speaker, were compared with all the voice signatures of the other speakers. Figure 33 shows each voice signature of Speaker 1 compared with all the voice signatures of other speakers.

For example the first voice signature of Speaker 1 compared to the first voice signature of Speaker 2, Speaker 3, Speaker 4, Speaker 5 and Speaker 6 is presented as $img1Sp1 \times img1$, each bar depicts one speaker.

Then the first voice signature of Speaker 1 was compared to the second, third and fourth voice signatures of other speakers $img1Sp1 \times img2$, $img1Sp1 \times img3$ and $img1Sp1 \times img4$.

The same was done to the second, third and fourth voice signatures of Speaker 1.

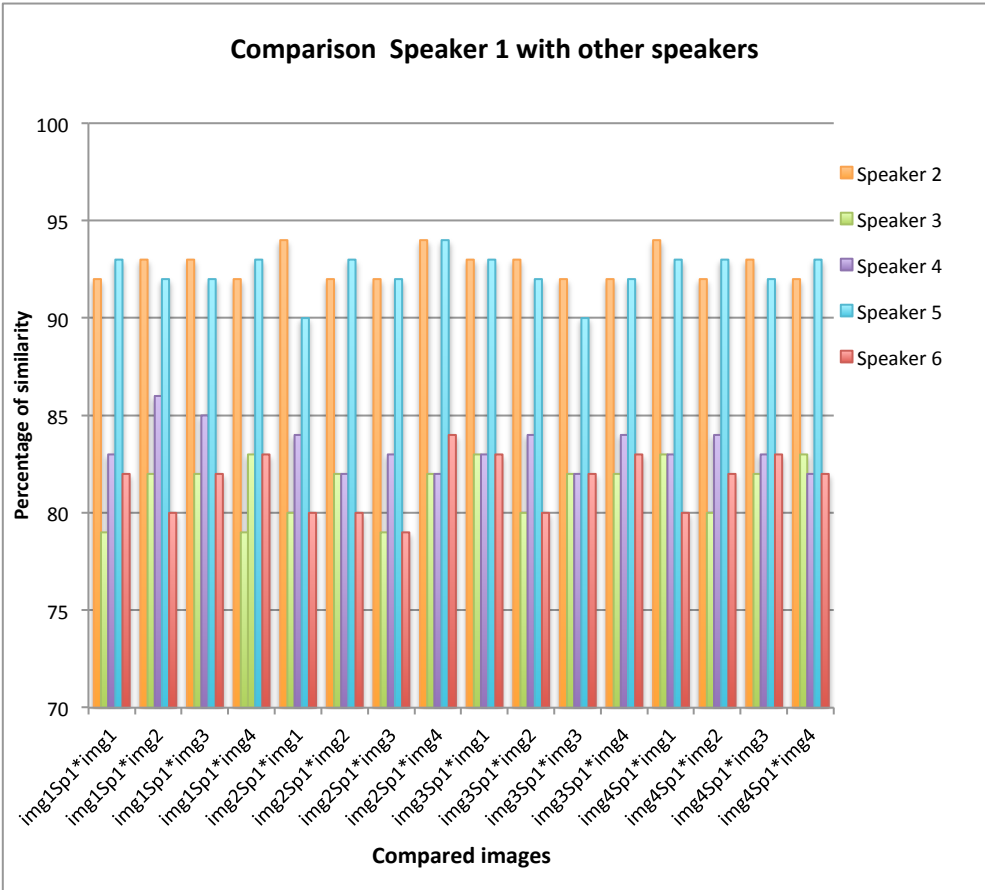


Figure 33: Comparisons voice images of Speaker 1 with the other speakers.

The similarity between the voice images range between 79% and 94%. Speakers 1, 2 and 5 were all females. For the same gender speakers—Speaker 1 compared to Speaker 2 and Speaker 5—the percentage range was between 90% and 94%. For different gender speakers—Speaker 1 compared to Speaker 3, Speaker 4 and Speaker 6—the similarity was between 79% and 86%.

According to the literature the frequency values of male individuals are lower than that of females, where the values range between 85Hz and 180Hz for males and between 165Hz and 255Hz for females (Latinus & Belin, 2011). That may be the reason behind the higher similarity for speakers of the same gender.

As a result it can be seen that it is difficult to distinguish between speakers by using this method because of the high rates of similarity between the voice images of different speakers.

Experiment 3

In this experiment, the four voice images of Speaker 3, a male speaker, were compared with the voice images of each speaker. In this experiment the male speaker was selected to compare with other speakers to see if it is possible to distinguish between two speakers using voice images, and how the similarities differ in the case of speakers with the same gender.

For example the first voice image of Speaker 3 was compared to the first voice image of Speaker 1, Speaker 2, Speaker 4, Speaker 5 and Speaker 6. This is represented as $img1Sp3 \times img1$, and each bar depicts one speaker.

Then the first voice image of Speaker 3 was compared with the second, third and fourth voice signatures of other speakers $img1Sp3 \times img2$, $img3Sp1 \times img3$ and $img1Sp3 \times img4$.

The same was done for the second, third and fourth voice images of Speaker 1. Figure 34 shows the results obtained from this experiment.

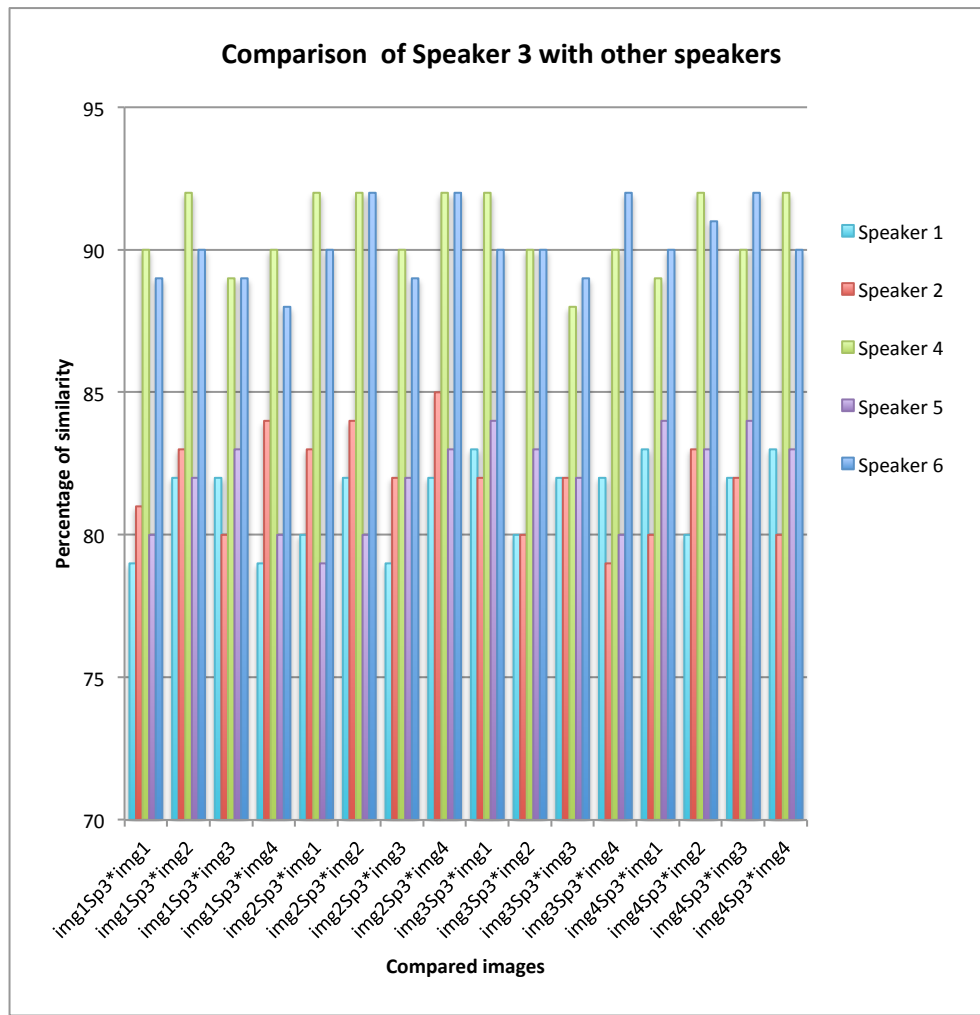


Figure 34: Comparisons of voice images of Speaker 3 with the other speakers.

Figure 34 depicts the comparisons of Speaker 3 to all the other speakers. As can be seen, the range of similarity in this case was between 79% and 92%. For the same gender speakers—Speaker 4 and Speaker 6—it was between 82% and 92%. For different gender speakers—Speaker 1, Speaker 2 and Speaker 5—it was between 79% and 84%.

The results of the three experiments of Phase 1 were *not satisfactory* because of the high level of similarity between the different speakers, especially when they are of the same gender. Thus this method is not accurate enough to use to identify speakers uniquely, and thus the experiments using visual representations were not extended.

4.2 Phase 2: MFCC Feature Extraction Algorithm with SVM classification.

During the second phase—the second DSR cycle—see Figure 35, the MFCC algorithm and the SVM were used to do the experiments. The TIMIT database was used for both training and testing.

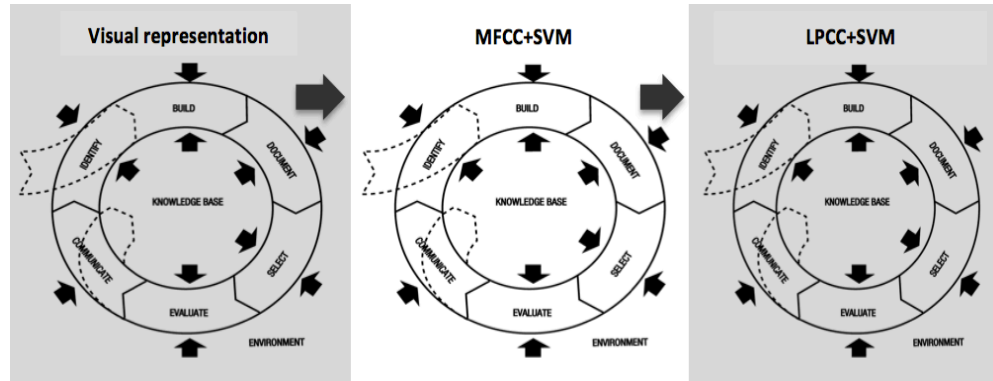


Figure 35: The second DSR cycle consisting of the MFCC+SVM phase.

Table 2 shows the main results of false acceptance, false rejection, true acceptance, and true rejection for each of the 35 speakers who were used to test the method of using MFCC combined with SVM.

Table 2: The main results obtained using MFCC combined with SVM.

Speaker	False acceptance	False rejection	True acceptance	True rejection	Accuracy
Sp1	1	1	3	3	75%
Sp2	0	1	4	3	87.5%
Sp3	1	0	3	4	87.5%
Sp4	1	1	3	3	75%
Sp5	0	1	4	3	87.5%
Sp6	1	1	3	3	75%
Sp7	0	1	4	3	87.5%
Sp8	1	1	3	3	75%
Sp9	0	1	4	3	87.5%
Sp10	0	0	4	4	100%
Sp11	0	1	4	3	87.5%
Sp12	1	1	3	3	75%
Sp13	0	1	4	3	87.5%

Speaker	False acceptance	False rejection	True acceptance	True rejection	Accuracy
Sp14	0	2	4	2	75%
Sp15	0	1	4	3	87.5%
Sp16	0	1	4	3	87.5%
Sp17	1	0	3	4	87.5%
Sp18	0	0	4	4	100%
Sp19	0	2	4	2	75%
Sp20	0	1	4	3	87.5%
Sp21	1	0	3	4	87.5%
Sp22	1	1	3	3	75%
Sp23	0	1	4	3	87.5%
Sp24	0	1	4	3	87.5%
Sp25	0	1	4	3	87.5%
Sp26	0	2	4	2	75%
Sp27	0	0	4	4	100%
Sp28	1	1	3	3	75%
Sp29	1	1	3	3	75%
Sp30	0	0	4	4	100%
Sp31	1	1	3	3	75%
Sp32	0	1	4	3	87.5%
Sp33	0	2	4	2	75%
Sp34	0	1	4	3	87.5%
Sp35	0	1	4	3	87.5%

According to the data in Table 2 the average accuracy for using this method is 84.29% with a standard deviation of 8.22%. Figure 36 depicts the recognition accuracy for each speaker of 35 speakers.

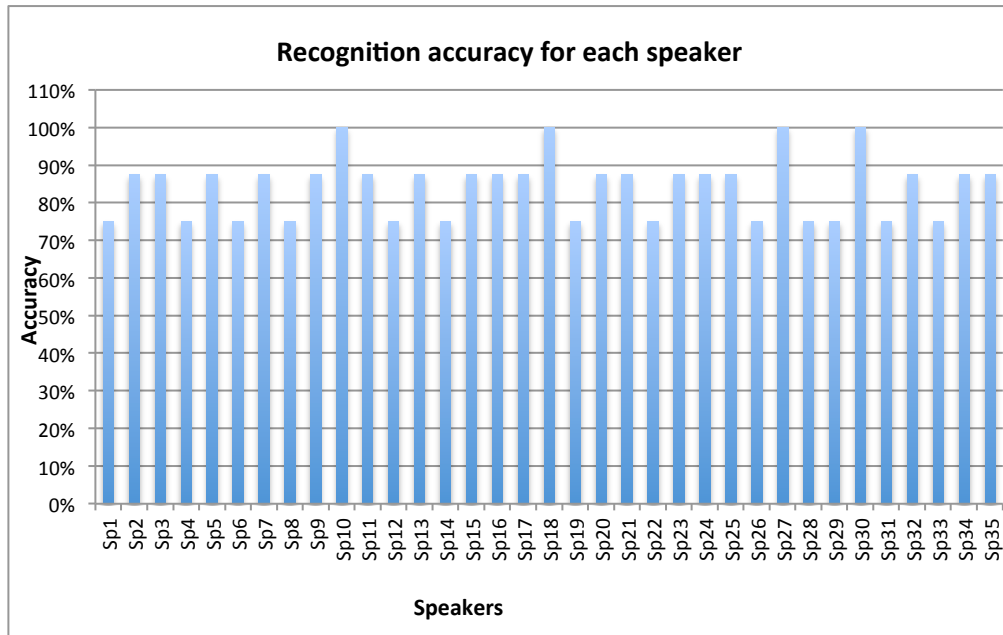


Figure 36: The accuracy of recognition for the 35 speakers using MFCC and SVM.

4.3 Phase 3: LPCC Feature Extraction Algorithm with SVM classification.

During the last phase—the third DSR cycle—see Figure 37, the LPCC algorithm was used to extract the features from the voice signal, and the SVM used these features for classification. The TIMIT database was used for both training and testing.

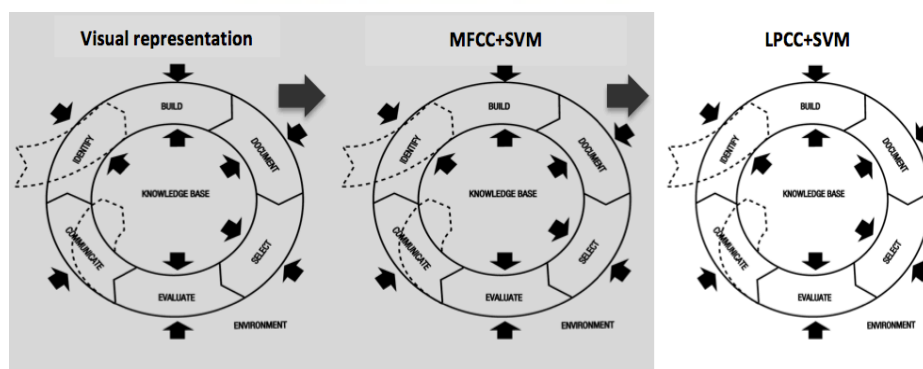


Figure 37: The third DSR cycle consisting of the LPCC+SVM phase.

Table 3 shows the main results obtained using the LPCC algorithm to extract the features of the voice signals and the SVM that was applied for classification.

Table 3: The main results obtained using LPCC combined with SVM.

Speaker	False acceptance	False rejection	True acceptance	True rejection	Accuracy
Sp1	1	1	3	3	75%
Sp2	0	1	4	3	87.5%
Sp3	1	1	3	3	75%
Sp4	1	1	3	3	75%
Sp5	1	1	3	3	75%
Sp6	0	1	4	3	87.5%
Sp7	0	1	4	3	87.5%
Sp8	1	1	3	3	75%
Sp9	0	1	4	3	87.5%
Sp10	0	2	4	2	75%
Sp11	0	1	4	3	87.5%
Sp12	1	1	3	3	75%
Sp13	0	1	4	3	87.5%
Sp14	0	2	4	2	75%
Sp15	0	2	4	2	75%
Sp16	0	1	4	3	87.5%
Sp17	1	0	3	4	87.5%
Sp18	0	0	4	4	100%
Sp19	1	1	3	3	75%
Sp20	1	1	3	3	75%
Sp21	0	1	4	3	87.5%
Sp22	1	1	3	3	75%
Sp23	0	1	4	3	87.5%
Sp24	0	0	4	4	100%
Sp25	0	1	4	3	87.5%
Sp26	0	2	4	2	75%
Sp27	0	0	4	4	100%
Sp28	1	1	3	3	75%
Sp29	1	1	3	3	75%
Sp30	0	0	4	4	100%
Sp31	1	1	3	3	75%
Sp32	0	3	4	1	87.5%

Speaker	False acceptance	False rejection	True acceptance	True rejection	Accuracy
Sp33	1	1	3	3	75%
Sp34	0	1	4	3	87.5%
Sp35	0	1	4	3	87.5%

According to the data in Table 3 the average accuracy of using LPCC combined with SVM to recognize the speaker is 82.9% with a standard deviation of 8.6%. Figure 38 represents the recognition accuracy of each one of the 35 speakers.

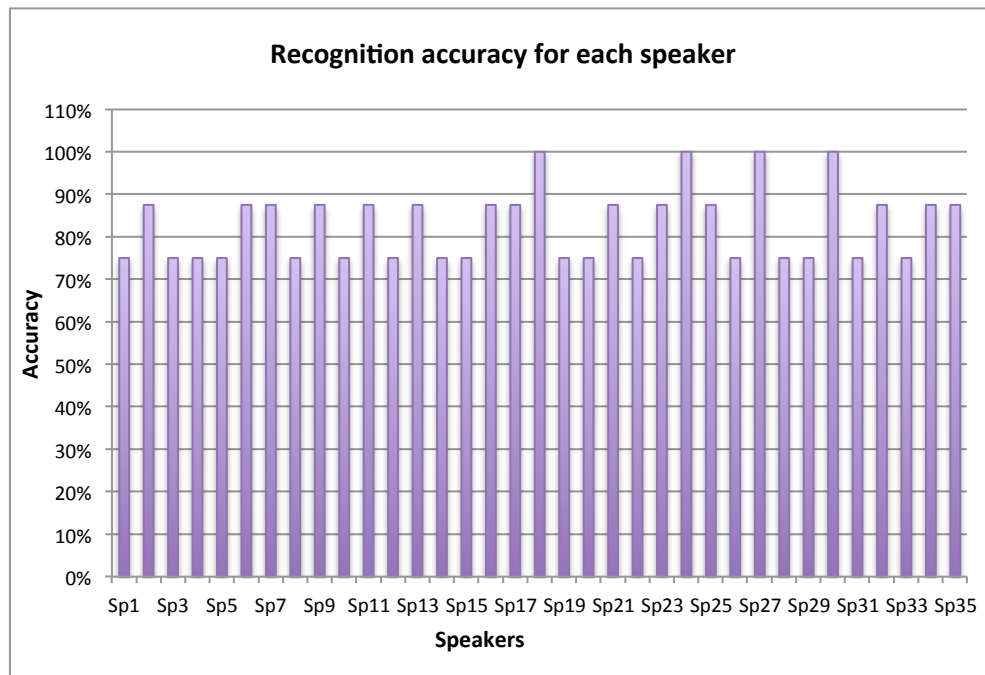


Figure 38: The accuracy of recognition for the 35 speakers using LPCC and SVM.

4.4 Comparison of Different Methods:

In the previous sections the main results were presented. The first method used the Jaccard similarity index to compare the voice images. This method yielded the poorest results.

The methods applied in Phases 2 and 3 gave encouraging results. However, using the MFCC algorithm with the SVM classifier, to recognise speakers, was found to be slightly more effective and accurate than using the LPCC algorithm with the SVM classifier.

Table 4 summaries the comparison of these two methods. It shows the average accuracy (ACC), false acceptance rate (FAR), and false rejection rate (FRR).

Where:

$$FAR = \frac{\text{Number of false acceptances}}{\text{Number of recognition attempts}}$$

and

$$FRR = \frac{\text{Number of false rejections}}{\text{Number of recognition attempts}}$$

Table 4: Comparison between MFCC+SVM and LPCC+SVM.

Methods	ACC	FAR	FRR
MFCC+SVM	84.3%	0.04	0.11
LPCC+SVM	82.9%	0.05	0.12

Since the data of the two methods is normally distributed with equal variances, the T-test will be reported, see Appendix C.

Table 5: Statistical comparison of the two methods

Methods	Mean	Standard Deviation	95% CI
MFCC+SVM	84.3%	8.22	81.5- 87.1
LPCC+SVM	82.5%	8.13	79.7-85.3

Since the 95% confidence intervals (CI) overlap, there is no significant difference between the two groups—MFCC+SVM and LPCC+SVM. To confirm this, the equal variance T-test = 0.9138 and the probability value = 0.3641, also indicate no significant difference between the two groups at a 5% level of significance, see Appendix C.

4.5 Summary

This chapter discussed the results obtained using the DSR methodology to do the experiments. These results were summarized and compared. Using the MFCC feature extraction algorithm and SVM for feature matching gave a slightly better result than LPCC+SVM. However, the difference has been shown not to be statistically significant.

In the next chapter the results will be discussed and some recommendations for future work will be given.



Chapter 5 Discussion and Conclusion

The aim of this research was to recognize a user through his voice and to identify the user by verifying that his/her voice is the same as a record of the voice signature. The research question addressed in this investigation was “What is the best way to identify a person using his / her voice signature?”

This overarching question was unpacked into the following three sub-questions:

- What feature extraction methods should be used?
- What classification method would be effective?
- What combinations of methods are reliable to authenticate a person’s voice signature?

In the previous chapter the results obtained, using various experiments, were presented. Several experiments were conducted using three different methods, to recognize users by means of voice signatures. These methods included: Jaccard similarity of linear frequency histograms of voice signatures, the MFCC feature extraction algorithm with SVM for speaker classification, and LPCC for feature extraction with SVM for speaker classification. The results achieved for each method were given. In this chapter, these findings will be discussed in terms of the research questions. In addition some suggestions will be made about how to improve voice-based speaker recognition.

5.1 Findings in Terms of the Research Questions

What is the best way to identify a person using his / her voice signature?

The methods that were used to address the first question were:

- Jaccard’s similarity method—it involved comparing the linear frequency histograms of each user’s voice image with the images of the other users’ voices;
- the mel frequency cepstrum coefficient feature extraction method was used to extract the unique features from the individual’s voice signal combined with a Support Vector Machine for classification.

- the linear prediction cepstral coefficient feature extraction method was used to extract the unique features from the individual's voice signature and classified by a Support Vector Machine trained on these extracted features.

The results achieved, using each of the methods, were compared to see which performs the best. The results of the analysis using the Jaccard similarity index method were found to be unsatisfactory. It was established that extracting the features from an individual's voice signal using a feature extraction method combined with a support vector machine for classification is the best way to identify a person.

What feature extraction methods should be used?

For the first sub question two different feature extraction algorithms were considered to achieve the aim of this research, the *mel frequency cepstrum coefficient* is the most popular feature extraction algorithm, which is based on a human being's hearing perceptions, and the *linear prediction cepstral coefficient*, which is based on the prediction of current voice samples as a linear combination of past voice samples.

According to the results that were presented in Chapter 4, both of these algorithms gave similar results with only a slight difference in accuracy. However, using the features extracted with the MFCC method gave an accuracy of 84.3% while feature extraction by LPCC gave an accuracy of 82.9%. This difference was shown not to be statistically significant.

What classification methods would be more effective?

Many feature classification methods exist to use for speaker recognition, however, according to the literature, the Support Vector Machine is one of the most popular feature classification methods for text-dependent speaker recognition (Salna & Kamarauskas, 2015) (Loh & Abdul Manan, 2010), and therefore this method was the one used for classification.

What combinations of methods are reliable to authenticate a person's voice signature?

For the last research question combinations of feature extraction and feature matching were used: firstly the MFCC feature extraction algorithm was used combined with SVM for

feature matching. Secondly the LPCC feature extraction algorithm was used combined with SVM for feature matching.

5.2 Difficulties Encountered

Some problems encountered during the study include:

- Data collection of the audio data for training and testing was difficult. In particular it was difficult to create consistent conditions for the recording of all the voice signatures. The need to collect a large corpus of data to improve training was also an impediment. Consequently, the TIMIT data set was considered instead of creating a self-generated data set. This TIMIT data set was available on-line and could be downloaded free of charge.
- Matlab was used to implement both feature extraction algorithms. Octave was also considered but it was found that MatLab has a richer machine learning toolkit.

5.3 Future Work

In future research the text-dependent speaker recognition system could be improved. The following are some suggestions to extend and improve the speaker recognition system.

Testing the system within different environments: for example in a noisy environment; when the speaker is using a phone; when a speaker is angry or sad, etc. and then to establish which method is more effective. To improve the system so that it is a text-independent speaker recognition system. That is, to improve the system so that it is able to identify the speaker with any spoken phrase. More feature extraction algorithms could be tested and compared to decide which is more effective when considering text-independent speaker recognition in different environments.

Many feature matching methods could be considered: for example, vector quantization, Gaussian mixture model, artificial neural network and hidden Markov model for text-dependent speaker recognition, and hidden Markov model, dynamic time warping for text-independent speaker recognition. Other combinations of methods could be used and could be compared to see which is the most effective and accurate method for recognizing a speaker's voice.

5.4 Conclusion

Reflecting on the study it is noted that there is currently a new interest in voice signatures, with for example Vodacom considering using voice signatures instead of a list of questions to identify their customers (www.vodacom.co.za/vodacom/services/internet/voice-password) this study therefore is relevant.



UNIVERSITY *of the*
WESTERN CAPE

Bibliography

- Achmed, I. (2014). *Independent hand-tracking from a single two-dimensional view and its application to South African sign language recognition*. University of the Western Cape, Computer Science, Cape Town.
- Alam, M., Kinnunen, T., Kenny, P., Ouellet, P., & O'Shaughnessy, D. (2013). Multitaper MFCC and PLP features for speaker verification using i-vectors. *Speech Communication*, 55 (2), 237–251.
- Bhattacharjee, U. (2013). A comparative study of LPCC and MFCC features for the recognition of Assamese phonemes. *International Journal of Engineering Research and Technology*, 2 (1), 1–6.
- Begum, N., Ulanova, L., Wang, J., & Keogh, E. (2015, August). Accelerating dynamic time warping clustering with a novel admissible pruning strategy. *In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 19–58. ACM.
- Beigi, H. (2011). *Fundamentals of Speaker Recognition*. Springer Science & Business Media.
- Bharti, R., & Bansal, P. (2015). Real time speaker recognition system using MFCC and vector quantization technique. *International Journal of Computer Applications*, 117 (1).
- Biggs, M., & Buchler, D. (2008). Eight criteria for practice-based research in the creative and cultural industries. *Art, Design & Communication in Higher Education*, 7 (1), 5–18.
- Bonneville, R., & Jin, V. (2013). A hidden Markov model to identify combinatorial epigenetic regulation for estrogen receptor target genes. *Bioinformatics*, 29 (1), 22–28.
- Borde, P., Varpe, A., Manze, R., & Yannawar, P. (2015). Recognition of isolated words using Zernike and MFCC features for audio visual speech recognition. *International Journal of Speech Technology*, 18 (2), 167–175.
- Boukhonine, S., Krotov, V., & Rupert, B. (2005). Future security approaches. *Communications of the Association for Information Systems*, 16 (1), 937–966.
- Bragagnini, A., Della Luna, S., Nocentini, S., & Turolla, M. S. (2014). *Patent No. U.S. Patent No. 8,665,062*. United States of America.
- Caudill, M. (1989). *Neural Network Primer Part I. AI Expert*.
- Celebi, S., Aydin, A., Temiz, T., & Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. *In VISAPP*, 620–625.
- Chakraborty, K., Talele, A., & Upadhyaya, S. (2014). Voice recognition using MFCC algorithm. *International Journal of Innovation Research in Advanced Engineer(IJIRAE) ISSN 2349-2163*, 1 (10), 158–161.
- Chapaneri, S. (2012). Spoken digits recognition using weighted MFCC and improve feature for dynamic time warping. *International Journal of Computer Applications*, 40 (3), 6–12.
- Christensson, P. (2006, Jun 1). *Waveform Definition*. Retrieved 2017.

- Collis, J., & Hussey, R. (2013). *Business Research: A Practical Guide for Undergraduate and Postgraduate Students*. Palgrave Macmillan.
- Crotty. (1998). *The Foundations of Social Research: Meaning and Perspective in the Research Process*. Sage Publications Ltd.
- Dave, N. (2013). Feature extraction methods LPC, PLP and MFCC in speech recognition. *International Journal for Advance Research in Engineering and Technology*, 1 (6), 1–4.
- Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustic, Speech and Signal Processing, IEEE Transactions*, 28 (4), 357–366.
- Feast, L. (2010, June). Epistemological position informing theories of design research: implications for the design discipline and design practice. *In Proceedings of Design and Complexity the 2010 Design Research Society International Conference*.
- Flood, A. (2010). Understanding phenomenology: Anne Flood looks at the theory and methods involved in phenomenological research. *Nurse Research*, 17 (2), 7–15.
- Flowers, P. (2009). *Research Philosophies—Important and Relevance*. Granfield School of Management.
- Ghahramani, Z. (2001). An Introduction to Hidden Markov models and Bayesian network. *Journal of Pattern Recognition and Artificial Intelligence*, 15, 9–42.
- Gold, B., Morgan, N., Ellis, D., & O'Shaughnessy, D. (2012). Speech and Audio Signal Processing: Processing and Perception of Speech and Music. *Journal of the Acoustical Society of America*, 132 (3), 1861.
- Grutzmacher, F., Wolff, J., & Haubelt, C. (2015). Exploiting the read-level parallelism in template-based gesture recognition with dynamic time warping. *In Proceedings of the 2nd International Workshop on Sensor-based Activity Recognition and Interaction* (p. 6). ACM.
- Gulzar, T., Singh, A., & Sharma, S. (2014). Comparative analysis of LPCC, MFCC and BFCC for the recognition of Hindi words using artificial neural networks. *International Journal of Computer Application*, 101 (12), 22–27.
- Harmansky, H. (1990). Perceptual linear predictive (PLP) Analysis of Speech. *The Journal of the Acoustical Society of America*, 87, 1738–1752.
- Hasan, M., Jamil, M., & Rahman, M. (2004). Speaker identification using mel frequency cepstral coefficients. *3rd International Conference on Electrical & Computer Engineering*, 1, pp. 565–568.
- Hastie, R., Tibshirani, & Friedman, J. (2009). *The Elements of Statistical Learning*. New York: Springer.
- Hearst, M., Dumais, S., Osman, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems*, 13 (4), 18–28.
- Huang, G., Chen, H., Zhou, Z., & Yin and K. Guo, F. (2011). Two-class support vector data description. *Pattern Recognition*, 44, 320–329.
- Huang, X., Acero, A., Hon, H., & Foreword By-Reddy, R. (2001). *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall.
- Jain, A. K., Flynn, P., & Ross, A. A. (2007). *Handbook of Biometrics*. Springer.

- Jain, A. K., Nandkumar, K., & Nagar, A. (2008). Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008 (113), 17.
- Jain, A., & Sharma, O. (2013). A vector quantization approach for voice recognition using mel frequency cepstral coefficients (MFCC): A review. *International Journal of Electronics & Communication Technology*, 26–29.
- Jain, A., Hong, L., & Pankanti, S. (2000). Biometrics of identification. *Communications of the ACM*, 43 (2), 91–98.
- Janse, P., Magre, S., Kurzekar, P., & Deshmukh, R. (2014). A comparative study between MFCC and DWT feature extraction technique. *International Journal of Engineering Research and Technology*, 3 (1), 3124–3127.
- Joshi, D., & Zalte, M. (2013). Joshi, D.D. and Zalte, M.B., 2013. Recognition of Emotion from Marathi Speech using MFCC and DWT algorithms. *International Journal of Advanced Computer Engineering and Communication Technology*, 2.
- Kamarauskas, J. (2015). Speaker recognition using Gaussian mixture models. *Elektronika in Elektrotechnika*, 85 (5), 29–32.
- Kaur, M., & Kaur, S. (2016). Automatic Speaker Recognition System. *International Advanced Research Journal in Science, Engineering and Technology*, 3 (2), 2394-1588.
- Kinnunen, T., & Li, H. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52 (1), 12–40.
- Knapp, M. (2008). Nonverbal rhetoric. In *The International Encyclopedia of Communication*. Blackwell.
- Ku, N., Chang, S., & Hwang, S. (2014). Binary search Vvctor quantization. *AASRI Procedia*, 8, 112–117.
- Latinus, M., & Belin, P. (2011). Human voice perception. *Current Biology*, 21 (4), R143–R145.
- Lee, C.-H., Soong, F. K., Paliwal, K., & (Eds). (2012). *Automatic speech and speaker recognition: advanced topics* (Vol. 355). Springer Science & Business Media.
- Loh, M. Y., & Abdul Manan, A. (2010). *Comparative study of speaker recognition methods: DTW, GMM and SVM*. University of Technology Malaysia, Faculty of Computer Science & Information System.
- Long, Y. (2006). *KernTune: Self-tuning linux kernel performance using support vector machines*. University of the Western Cape, Computer Science, Cape Town.
- Makhijani, R., & Gupta, R. (2013). Isolated word speech recognition system using dynamic time warping. *International Journal of Engineering Sciences & Emerging Technologies (IJESSET)*, 6 (3), 352-367.
- Mansour, A. H., Salh, G., & Mohammed, K. (2015). Voice Recognition using dynamic time warping and mel-frequency cepstral coefficients algorithms. *International Journal of Computer Applications*, 116 (2).
- Mishra, P. (2012). A vector quantization approach to speaker recognition. *Proceedings of International Conference on Innovation & Research in Technology for Sustainable Development*, (pp. 152–155).
- Monrose, Fabian, & Aviel, D. R. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems*, 16 (4), 351–359.

- Mordini, E., & Tzovaras, D. (2012). *Second Generation Biometrics: The Ethical, Legal and Social Context* (Vol. 11). Springer Science & Business Media.
- Moreno, P., & Ho, P. (2003). A new SVM approach to speaker identification and verification using probabilistic kernels. *Eurospeech*, (pp. 1–9).
- Muda, L., Begam, M., & Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *Journal of Computing*, 2 (3), 138–143.
- Murty, K., & Yegnanarayana, B. (2006). Combining evidence from residual phase and MFCC feature for speaker recognition. *Signal Processing Letters, IEEE*, 13 (1), 52–55.
- Nakagawa, S., Asakawa, K., & Wang, L. (2007). Speaker recognition by combining MFCC and phase information. *Interspeech*, 60, pp. 2005–2008.
- Nijhawan, G., & Soni, M. (2014). Speaker recognition using MFCC and vector quantization. *International Journal on Recent Trends in Engineering and Technology*, 11 (1), 211–218.
- Noble, W. (2006). What is a support vector machine? *Nature Biotechnology*, 24 (12), 1565–1567.
- Pal, S., Pal, U., & Blumenstein, M. (2014). *Signature-Based Biometric Authentication. In Computational Intelligence in Digital Forensics: Forensic Investigation and Applications*. Springer.
- Parul, R. (2012). Automatic Speaker Recognition System. *International Journal of Advanced Computer Research*, 2 (7), 2277-7970.
- Patel, J., & Nadurbarkar, A. (2015). Development and implementation of algorithm for speaker recognition for Gujarati language. *International Research Journal of Engineering and Technology*, 2 (2), 444–448.
- Patel, K., & Prasad, R. (2013). Speech recognition and verification using MFCC & VQ. *International Journal of Emerging Science and Engineering*, 1 (7), 33–37.
- Poyatos, F. (2002). *Nonverbal Communication Across Disciplines. Volume I: Culture, Sensory Interaction, Speech, Conversation*.
- Prabhakar, S., Pankanti, S., & Jain, A. K. (2003). Biometric recognition : Security and privacy concerns. *IEEE Security & Privacy*, 1 (2), 33–42.
- Prakash, S., & Gupta, P. (2015). Introduction. *Ear Biometric in 2D and 3D*, (pp. 1–20). Singapore.
- Rabiner, L., & Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice Hall, Inc., Englewood Cliffs, New Jersey .
- Rachna, X., Singh, D., & Vikas, Y. (2014). Feature extraction from asthma patient's voice using mel-frequency cepstral coefficients. *International Journal of Research in Engineering and Technology*, 3 (6), 2321-7308,.
- Rajasekar, S., Philomonathan, P., & Chinnathambi, V. (2006). *Research Methodology*. arXiv.
- Reynolds, D. (2002). An overview of automatic speaker recognition. *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)(S.4072-4075)*.
- Rubin, H., & Rubin, I. (2012). Research philosophy and qualitative interviews. *In H. J. Rubin & I. S. Rubin, Qualitative Interviewing: The Art of Hearing Data.*, 13–24.

- Saini, R., & Rana, N. (2014). Comparison of various biometric methods. *International Journal of Advances in Science and Technology*, 2 (1), 2.
- Saleh, M., Ibrahim, N., & Ramli, D. (2014). Data reduction on MFCC features based on kernel PCA for speaker verification system. *WALIA journal*, 30 (2), 56–62.
- Salna, B., & Kamarauskas, J. (2015). Evaluation of effectiveness of different methods in speaker recognition. *Elektronika in Elektrotechnika*, 98 (2), 67–70.
- Scholkopf, B., Mika, S., Burges, C., Knirsch, P., Muller, K., Ratsch, G., et al. (1999). Input space versus feature space in kernel-based methods. *Neural Networks, IEEE Trans*, 10 (5), 1000–1017.
- Shah, F. (2009). Automatic emotion recognition from speech using artificial neural network with gender-dependent database. In *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies. IEEE*, 162–164.
- Silveira, M., Schroeder, C., da Costa, J., de Oliveira, C., Junior, J., & Junior, S. (2013). Convolutional ICA-based forensic speaker identification using mel frequency cepstral coefficients and Gaussian mixture models. *The International Journal of Forensic Computer Science*, 1, 27–34.
- Sirvastava, N. (2014). Speech recognition using artificial neural networks. *International Journal of Environmental Science and Technology*, 3 (3).
- Song, S. (2015). *African Undersea Cables*. From Flickr: <https://www.flickr.com/photos/ssong/17069911790>
- Srinivasan, A. (2012). Speaker identification and verification using vector quantization and mel frequency cepstral coefficients. *Research Journal of Applied Sciences*, 4 (1), 33–40.
- Sutharan, S. (2016). Support vector machines. In *Machine Learning Models and Algorithms for Big Data Classification* (pp. 207–235). Springer.
- Swaminathan, S., & Jayasankar, T. (2015). Emotion speech recognition using MFCC and residual phase in artificial neural network. *Int. J. Engg. Res. & Sci. & Tech*, 4 (3), 106–113.
- Tallat, S., Shahid, F., Samad, A., & Abbasi, M. U. (2014). *Voice identification and recognition system*. Islamabad: Department of Electrical Engineering, COMSATS.
- Tarango, J., Keogh, E., & Brisk, P. (2014). Accelerating the dynamic time wrapping distance measure using logarithmical arithmetic. *ACSSC*, 404–408.
- Tiwari, V. (2010). MFCC and its applications in speaker recognition. *International Journal on Emerging Technologies*, 1 (1), 19–22.
- Tzotsos, A. (2006). A support vector machine approach for object based image analysis. *Proceedings of the First International Conference on Object-Based Image Analysis*, (pp. 1–6).
- Utane, A., & Nalbawar, S. (2013). Emotion recognition through speech using Gaussian mixture model and support vector machine. *International Journal of Scientific & Engineering Research*, 2 (8), 1439–1443.
- Veton Z, K., & Hussien A., E. (2015). Robust speech recognition system using conventional and hybrid features of MFCC, LPCC, PLP, RASTA-PLP and hidden Markov model classifier in noisy conditions. *Journal of Computer and Communications*, 3, 1–9.

- Vladutescu, S. (2013). Seven qualitative-soft communicative characteristics of human voice. *Journalism and Mass Communication*, 3 (10), 103–107.
- Vladutescu, S. (2013). Two quantitative-hard communicative characteristics of human voice. *European Journal of Business and Social Sciences*, 2 (6), 103–107.
- Vom Brocke, J., & Buddendick, C. (2006). Reusable conceptual models–requirements based on the design science research paradigm. *Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*, (pp. 576–604). Claremont.
- Weston, J. (2014). *Support Vector Machine*. From NEC Labs America: http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf
- Whitehill, J. (2006). *Automatic real-time facial expression recognition for signed language translation*. University of the Western Cape, Computer Science, Cape Town.
- Xu, H. H. (2015). Text dependent speaker recognition study. *International Conference on Advances in Mechanical Engineering and Industrial Informatics* (pp. 136–139). Atlantis Press.
- Yang, S., Cao, J., & Wang, J. (2015). Acoustics recognition of construction equipment based on LPCC features and SVM. *34th Chinese Control Conference*, 3987–3991.
- Yildirim, S., Singh, S., Dean, T., & Jasra, A. (2015). Parameter estimation in hidden Markov models with intractable likelihoods using sequential Monte Carlo. *Journal of Computational and Graphical Statistics*, 24 (3), 846–865.
- Zhang, X., Sun, J., & Luo, Z. (2014). One-against all weighted dynamic time warping for language-independent and speaker-dependent Speech Recognition in Adverse Conditions. *PLOS ONE*, 9 (2), 1–9.
- Zilli, D., Parson, O., Merrett, G., & Rogers, A. (2014). A hidden Markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring. *Journal of Artificial Intelligence Research*, 805–827.

Appendices

Appendix A: TIMIT data base loading

```
function [this default_path] = ADT( dbName ,init_dir,flags)
%ADT - constructs a MatlabADT object.
%Technion SIPL MatlabADT (Audio Database Toolbox)
%Implemented by: Kobi Nistel
%Supervised by: Yevgeni Litvin and Yair Moshe
%Version 1.01, Jan 2009
%Technical support: matlab_adt@sipl.technion.ac.il
%Lab site: www-sipl.technion.ac.il
%
%[ADTobj default_path] = ADT(dbName,init_dir,flags)
%All operations on the database will be performed using the
%ADT object which is passed to them as the first parameter.
%Exemples:
% db = ADT; - loads TIMIT database form defalut path.
% db2 = ADT('ctimit'); - loads CTIMIT database form the defalut path.
%Setup:
% on operating MatlabADT outside of SIPL for the first time run the command
% db = ADT('timit','c:\timitPath','setup');
%
%See also query, filterdb, read, play.

%checks the location of the timitdb directory
data_file_path = which('@ADT/ADT'); %to be fixed
data_file_path = data_file_path(1:end-5); %to be fixed

if (nargin<1)
    dbName = 'timit';
end
if (nargin<2)
    init_dir = textread([data_file_path dbName '_path.txt'],'%g');
    init_dir = init_dir{1};
end
if (nargin<3)
    flags='non';
end
if(strcmpi(flags,'setPath') || strcmpi(flags,'setup') ||
strcmpi(flags,'reBuild '))
    fid = fopen([data_file_path dbName '_path.txt'],'w+');
    fwrite(fid,[''' init_dir ''']);
    fclose(fid);
end

persistent cachedDBname;
persistent cachedDataBase; %for faster loading
this = databaseclass;
this.path = init_dir; %sets global path
this.kind = 1; %sets DB as sentence
this.name = dbName; %to do:from a file

%checks for exisistence of datafile
if( ~exist([data_file_path ,dbName,'.mat'], 'file') || strcmpi(flags,'reBuild
'))
    this = makedb(init_dir,this);
    save([data_file_path,dbName],'this');
    cachedDataBase = this;
else
```

```

fprintf('Loading %s...\n',this.name);
if(isempty(cachedDataBase) || ~strcmp(cachedDBname,dbName))
    load([data_file_path,dbName]);
    this.path = init_dir; %sets db path
    cachedDataBase = this;
else
    this = cachedDataBase;
    this.path = init_dir; %sets db path
end
end
default_path = this.path;
fprintf('Enteries: %d\n',this.entriesNumber);
cachedDBname = dbName;
if strcmpi(flags,'setup')
    play(this,1);
end
end

function this = makedb(init_dir,this)
%MAKEDB - generates the database file
fprintf('Generating database file:');
for train_testC=1:2 %USAGE: Test / Run
    switch train_testC
        case 1, train_test = 'train';
        case 2, train_test = 'test';
    end
    for dialectC=1:8 % Dialects
        fprintf('.');
        switch dialectC
            case 1, dialect = 'dr1';
            case 2, dialect = 'dr2';
            case 3, dialect = 'dr3';
            case 4, dialect = 'dr4';
            case 5, dialect = 'dr5';
            case 6, dialect = 'dr6';
            case 7, dialect = 'dr7';
            case 8, dialect = 'dr8';
        end
        speaker_dirs = dir([init_dir,'\',train_test,'\',dialect,'\*.']);
        %speakers
        for sex_speakerC = 1:length(speaker_dirs)
            if( strcmpi(speaker_dirs(sex_speakerC).name(1) , 'F') )
                sex = 'F';
            else
                sex = 'M';
            end
            speaker = speaker_dirs(sex_speakerC).name(2:end); %Cutting the
F/M
        direct=[init_dir,'\',train_test,'\',dialect,'\',speaker_dirs(sex_speakerC).na
me];
        files=dir([direct,'\*.wav']);
        for sentenceC = 1:length(files)
            %read on sentence data:
            this.entries(end+1).ID = length(this.entries); %couses an
empty cell in db(1)!!
            this.entries(end).sentence = files(sentenceC).name(1:end-4);
            %no exonetion
            this.entries(end).usage = train_test;
            this.entries(end).dialect = dialect;
            this.entries(end).sex = sex;
            this.entries(end).speaker = speaker;
        end
    end
end

```


Appendix B: MFCC algorithm

```
function [ CC, FBE, frames ] = mfcc( speech, fs, Tw, Ts, alpha, window, R, M,  
N, L )  
% MFCC Mel frequency cepstral coefficient feature extraction.  
%  
% MFCC(S,FS,TW,TS,ALPHA,WINDOW,R,M,N,L) returns mel frequency  
% cepstral coefficients (MFCCs) computed from speech signal given  
% in vector S and sampled at FS (Hz). The speech signal is first  
% preemphasised using a first order FIR filter with preemphasis  
% coefficient ALPHA. The preemphasised speech signal is subjected  
% to the short-time Fourier transform analysis with frame durations  
% of TW (ms), frame shifts of TS (ms) and analysis window function  
% given as a function handle in WINDOW. This is followed by magnitude  
% spectrum computation followed by filterbank design with M triangular  
% filters uniformly spaced on the mel scale between lower and upper  
% frequency limits given in R (Hz). The filterbank is applied to  
% the magnitude spectrum values to produce filterbank energies (FBEs)  
% (M per frame). Log-compressed FBEs are then decorrelated using the  
% discrete cosine transform to produce cepstral coefficients. Final  
% step applies sinusoidal lifter to produce liftered MFCCs that  
% closely match those produced by HTK [1].  
%  
% [CC,FBE,FRAMES]=MFCC(...) also returns FBEs and windowed frames,  
% with feature vectors and frames as columns.  
%  
% This framework is based on Dan Ellis' rastamat routines [2]. The  
% emphasis is placed on closely matching MFCCs produced by HTK [1]  
% (refer to p.337 of [1] for HTK's defaults) with simplicity and  
% compactness as main considerations, but at a cost of reduced  
% flexibility. This routine is meant to be easy to extend, and as  
% a starting point for work with cepstral coefficients in MATLAB.  
% The triangular filterbank equations are given in [3].  
%  
% Inputs  
% S is the input speech signal (as vector)  
%  
% FS is the sampling frequency (Hz)  
%  
% TW is the analysis frame duration (ms)  
%  
% TS is the analysis frame shift (ms)  
%  
% ALPHA is the preemphasis coefficient  
%  
% WINDOW is a analysis window function handle  
%  
% R is the frequency range (Hz) for filterbank analysis  
%  
% M is the number of filterbank channels  
%  
% N is the number of cepstral coefficients  
% (including the 0th coefficient)  
%  
% L is the liftering parameter  
%  
% Outputs  
% CC is a matrix of mel frequency cepstral coefficients  
% (MFCCs) with feature vectors as columns  
%  
% FBE is a matrix of filterbank energies  
% with feature vectors as columns  
%  
% FRAMES is a matrix of windowed frames  
% (one frame per column)
```

```

%
% Example
%     Tw = 25;           % analysis frame duration (ms)
%     Ts = 10;          % analysis frame shift (ms)
%     alpha = 0.97;     % preemphasis coefficient
%     R = [ 300 3700 ]; % frequency range to consider
%     M = 20;           % number of filterbank channels
%     C = 13;           % number of cepstral coefficients
%     L = 22;           % cepstral sine lifter parameter
%
%     % hamming window (see Eq. (5.2) on p.73 of [1])
%     hamming = @(N)(0.54-0.46*cos(2*pi*[0:N-1]./(N-1)));
%
%     % Read speech samples, sampling rate and precision from file
%     [ speech, fs, nbits ] = wavread( 'sp10.wav' );
%
%     % Feature extraction (feature vectors as columns)
%     [ MFCCs, FBES, frames ] = ...
%         mfcc( speech, fs, Tw, Ts, alpha, hamming, R, M,
C, L );
%
%     % Plot cepstrum over time
%     figure('Position', [30 100 800 200], 'PaperPositionMode', 'auto',
...
%         'color', 'w', 'PaperOrientation', 'landscape', 'Visible',
'on' );
%
%     imagesc( [1:size(MFCCs,2)], [0:C-1], MFCCs );
%     axis( 'xy' );
%     xlabel( 'Frame index' );
%     ylabel( 'Cepstrum index' );
%     title( 'Mel frequency cepstrum' );
%
%
%% PRELIMINARIES

% Ensure correct number of inputs
if( nargin~= 10 ), help mfcc; return; end;

% Explode samples to the range of 16 bit shorts
if( max(abs(speech))<=1 ), speech = speech * 2^15; end;

Nw = round( 1E-3*Tw*fs ); % frame duration (samples)
Ns = round( 1E-3*Ts*fs ); % frame shift (samples)

nfft = 2^nextpow2( Nw ); % length of FFT analysis
K = nfft/2+1; % length of the unique part of the FFT

%% HANDY INLINE FUNCTION HANDLES

% Forward and backward mel frequency warping (see Eq. (5.13) on p.76 of
[1])
% Note that base 10 is used in [1], while base e is used here and in HTK
code
hz2mel = @( hz )( 1127*log(1+hz/700) ); % Hertz to mel warping
function
mel2hz = @( mel )( 700*exp(mel/1127)-700 ); % mel to Hertz warping
function

% Type III DCT matrix routine (see Eq. (5.14) on p.77 of [1])
dctm = @( N, M )( sqrt(2.0/M) * cos( repmat([0:N-1].',1,M) ...
.* repmat(pi*([1:M]-0.5)/M,N,1) ) );

```

```

% Cepstral lifter routine (see Eq. (5.12) on p.75 of [1])
ceplifter = @( N, L )( 1+0.5*L*sin(pi*[0:N-1]/L) );

%% FEATURE EXTRACTION

% Preemphasis filtering (see Eq. (5.1) on p.73 of [1])
speech = filter( [1 -alpha], 1, speech ); % fvtool( [1 -alpha], 1 );

% Framing and windowing (frames as columns)
frames = vec2frames( speech, Nw, Ns, 'cols', window, false );

% Magnitude spectrum computation (as column vectors)
MAG = abs( fft(frames,nfft,1) );

% Triangular filterbank with uniformly spaced filters on mel scale
H = trifbank( M, K, R, fs, hz2mel, mel2hz ); % size of H is M x K

% Filterbank application to unique part of the magnitude spectrum
FBE = H * MAG(1:K,:); % FBE( FBE<1.0 ) = 1.0; % apply mel floor

% DCT matrix computation
DCT = dctm( N, M );

% Conversion of logFBEs to cepstral coefficients through DCT
CC = DCT * log( FBE );

% Cepstral lifter computation
lifter = ceplifter( N, L );

% Cepstral liftering gives liftered cepstral coefficients
CC = diag( lifter ) * CC; % ~ HTK's MFCCs

% EOF

function FMatrix=kannumfcc(num,s,Fs)

n=512; %Number of FFT points
Tf=0.025; %Frame duration in seconds
N=Fs*Tf; %Number of samples per frame
fn=24; %Number of mel filters
l=length(s); %total number of samples in speech
Ts=0.01; %Frame step in seconds
FrameStep=Fs*Ts; %Frame step in samples
a=1;
b=[1, -0.97]; %a and b are high pass filter coefficients

noFrames=floor(l/FrameStep); %Maximum no of frames in speech sample
FMatrix=zeros(noFrames-2, num); %Matrix to hold cepstral coefficients
lifter=1:num; %Lifter vector index
lifter=1+floor((num)/2)*(sin(lifter*pi/num));%raised sine lifter version

if mean(abs(s)) > 0.01
    s=s/max(s); %Normalises to compensate for mic vol
differences
end

%Segment the signal into overlapping frames and compute MFCC coefficients
for i=1:noFrames-2
    frame=s((i-1)*FrameStep+1:(i-1)*FrameStep+N); %Holds individual frames
    Ce1=sum(frame.^2); %Frame energy
    Ce2=max(Ce1,2e-22); %floors to 2 X 10 raised to power -22
    Ce=log(Ce2);
    framef=filter(b,a,frame); %High pass pre-emphasis filter

```

```

    F=framef.*hamming(N);           %multiplies each frame with hamming window
    FFTo=fft(F,N);                  %computes the fft
    melf=melbankm(fn,n,Fs);        %creates 24 filter, mel filter bank
    halfn=1+floor(n/2);
    spectr1=log10(melf*abs(FTTo(1:halfn)).^2);%result is mel-scale filtered
    spectr=max(spectr1(:),1e-22);
    c=dct(spectr);                  %obtains DCT, changes to cepstral domain
    c(1)=Ce;                        %replaces first coefficient
    coeffs=c(1:num);                %retains first num coefficients
    ncoeffs=coeffs.*lifter';        %Multiplies coefficients by lifter value
    FMatrix(i, :)=ncoeffs';        %assigns mfcc coeffs to successive rows i
end
%% Train a Support Vector Machine Classifier
%%
% Load Fisher's iris data set. Remove the sepal lengths and widths, and all
% observed setosa irises.

% Copyright 2015 The MathWorks, Inc.

load fisheriris
inds = ~strcmp(species,'setosa');
X = meas(inds,3:4);
y = species(inds);
%%
% Train an SVM classifier using the processed data set.
SVMModel = fitcsvm(X,y)
%%
% The Command Window shows that |SVMModel| is a trained |ClassificationSVM|
% classifier and a property list. Display the
% properties of |SVMModel|, for example, to determine the class order, by
% using
% dot notation.
classOrder = SVMModel.ClassNames
%%
% The first class (|'versicolor'|) is the negative class, and the second
% (|'virginica'|) is the positive class. You can change the class order
% during training by using the '|ClassNames'| name-value pair argument.
%%
% Plot a scatter diagram of the data and circle the support vectors.
sv = SVMModel.SupportVectors;
figure
gscatter(X(:,1),X(:,2),y)
hold on
plot(sv(:,1),sv(:,2),'ko','MarkerSize',10)
legend('versicolor','virginica','Support Vector')
hold off
%%
% The support vectors are observations that occur on or beyond their
% estimated class boundaries.
%%
% You can adjust the boundaries (and therefore the number of support
% vectors) by setting a box constraint during training using the
% '|BoxConstraint'| name-value pair argument.
%% Train and Cross Validate an SVM Classifier
%%
% Load the |ionosphere| data set.

% Copyright 2015 The MathWorks, Inc.

load ionosphere
rng(1); % For reproducibility
%%
% Train an SVM classifier using the radial basis kernel. Let the software
% find a scale value for the kernel function. It is good
% practice to standardize the predictors.

```

```

SVMModel = fitcsvm(X,Y,'Standardize',true,'KernelFunction','RBF',...
'KernelScale','auto');
%%
% |SVMModel| is a trained |ClassificationSVM| classifier.
%%
% Cross validate the SVM classifier. By default, the software uses
% 10-fold cross validation.
CVSVMModel = crossval(SVMModel);
%%
% |CVSVMModel| is a |ClassificationPartitionedModel| cross-validated
%% Linear Prediction and Autoregressive Modeling
% This example shows how to compare the relationship between autoregressive
% modeling and linear prediction. Linear prediction and autoregressive
% modeling are two different problems that can yield the same numerical
% results. In both cases, the ultimate goal is to determine the parameters
% of a linear filter. However, the filter used in each problem is
% different.

%% Introduction
% In the case of linear prediction, the intention is to determine an FIR
% filter that can optimally predict future samples of an autoregressive
% process based on a linear combination of past samples. The difference
% between the actual autoregressive signal and the predicted signal is
% called the prediction error. Ideally, this error is white noise.
%
% For the case of autoregressive modeling, the intention is to determine an
% all-pole IIR filter, that when excited with white noise produces a signal
% with the same statistics as the autoregressive process that we are trying
% to model.

%% Generate an AR Signal using an All-Pole Filter with White Noise as Input
% Here we use the LPC function and an FIR filter simply to come up with
% parameters we will use to create the autoregressive signal we will work
% with. The use of FIR1 and LPC are not critical here. For example, we
% could replace d with something as simple as [1 1/2 1/3 1/4 1/5 1/6 1/7
% 1/8] and p0 with something like 1e-6. But the shape of this filter is
% nicer so we use it instead.
b = fir1(1024, .5);
[d,p0] = lpc(b,7);

%%
% To generate the autoregressive signal, we will excite an all-pole filter
% with white gaussian noise of variance p0. Notice that to get variance p0,
% we must use SQRT(p0) as the 'gain' term in the noise generator.
rng(0,'twister'); % Allow reproduction of exact experiment
u = sqrt(p0)*randn(8192,1); % White gaussian noise with variance p0

%%
% We now use the white gaussian noise signal and the all-pole filter to
% generate an AR signal.
x = filter(1,d,u);

%% Find AR Model from Signal using the Yule-Walker Method
% Solving the Yule-Walker equations, we can determine the parameters for an
% all-pole filter that when excited with white noise will produce an AR
% signal whose statistics match those of the given signal, x. Once again,
% this is called autoregressive modeling. In order to solve the Yule-Walker
% equations, it is necessary to estimate the autocorrelation function of x.
% The Levinson algorithm is used then to solve the Yule-Walker equations in
% an efficient manner. The function ARYULE does all this for us.
[d1,p1] = aryule(x,7);

%% Compare AR Model with AR Signal
% We now would like to compute the frequency response of the all-pole

```



```

% filter we have just used to model the AR signal x. It is well-known that
% the power spectral density of the output of this filter, when the filter
% is excited with white gaussian noise is given by the magnitude-squared of
% its frequency response multiplied by the variance of the white-noise
% input. One way to compute this output power spectral density is by using
% FREQZ as follows:
[H1,w1]=freqz(sqrt(p1),d1);

%%
% In order to get an idea of how well we have modeled the autoregressive
% signal x, we overlay the power spectral density of the output of the
% model, computed using FREQZ, with the power spectral density estimate of
% x, computed using PERIODOGRAM. Notice that the periodogram is scaled by
% 2*pi and is one-sided. We need to adjust for this in order to compare.
periodogram(x);
hold on;
hp = plot(w1/pi,20*log10(2*abs(H1)/(2*pi)), 'r'); % Scale to make one-sided
PSD
hp.LineWidth = 2;
xlabel('Normalized frequency (\times \pi rad/sample)')
ylabel('One-sided PSD (dB/rad/sample)')
legend('PSD estimate of x','PSD of model output')

%% Use LPC to Perform Linear Prediction
% We now turn to the linear prediction problem. Here we try to determine an
% FIR prediction filter. We use LPC to do so, but the result from LPC
% requires a little interpretation. LPC returns the coefficients of the
% entire whitening filter A(z), this filter takes as input the
% autoregressive signal x and returns as output the prediction error.
% However, A(z) has the prediction filter embedded in it, in the form B(z)
% = 1- A(z), where B(z) is the prediction filter. Note that the
% coefficients and error variance computed with LPC are essentially the
% same as those computed with ARYULE, but their interpretation is different.
[d2,p2] = lpc(x,7);
[d1.', d2.']

%%
% We now extract B(z) from A(z) as described above to use the FIR linear
% predictor filter to obtain an estimate of future values of the
% autoregressive signal based on linear combinations of past values.
xh=filter(-d2(2:end),1,x);

%% Compare Actual and Predicted Signals
% To get a feeling for what we have done with a 7-tap FIR prediction
% filter, we plot (200 samples) of the original autoregressive signal along
% with the signal estimate resulting from the linear predictor keeping in
% mind the one-sample delay in the prediction filter.
cla
stem([x(2:end),xh(1:end-1)]);
xlabel('Sample time');
ylabel('Signal value');
legend('Original autoregressive signal','Signal estimate from linear
predictor')
axis([0 200 -0.08 0.1])

%% Compare Prediction Errors
% The prediction error power (variance) is returned as the second output
% from LPC. Its value is (theoretically) the same as the variance of the
% white noise driving the all-pole filter in the AR modeling problem (p1).
% Another way of estimating this variance is from the prediction error
% itself:
p3 = norm(x(2:end)-xh(1:end-1),2)^2/(length(x)-1);

%%

```

```
% All of the following values are theoretically the same. The
% differences are due to the various computation and approximation errors
% herein.
[p0,p1,p2,p3]
```



UNIVERSITY *of the*
WESTERN CAPE

Appendix C

Two-Sample Test Report

Variable C1

Descriptive Statistics Section

Variable	Count	Mean	Standard Deviation	Standard Error	95.0% LCL of Mean	95.0% UCL of Mean
C2=1	35	84.28571	8.215199	1.388622	81.46369	87.10773
C2=2	35	82.5	8.134892	1.375048	79.70557	85.29443

Note: T-alpha (C2=1) = 2.0322, T-alpha (C2=2) = 2.0322

Confidence-Limits of Difference Section

Variance Assumption	DF	Mean Difference	Standard Deviation	Standard Error	95.0% LCL Difference	95.0% UCL Difference
Equal	68	1.785714	8.175144	1.954233	-2.113898	5.685326
Unequal	67.99	1.785714	11.5614	1.954233	-2.113904	5.685333

Note: T-alpha (Equal) = 1.9955, T-alpha (Unequal) = 1.9955

Equal-Variance T-Test Section

Alternative Hypothesis	T-Value	Probability Level	Reject H0 at .050	Power (Alpha=.050)	Power (Alpha=.010)
Difference <> 0	0.9138	0.364068	No	0.146900	0.046343
Difference < 0	0.9138	0.817966	No	0.005392	0.000636
Difference > 0	0.9138	0.182034	No	0.229600	0.076270

Aspin-Welch Unequal-Variance Test Section

Alternative Hypothesis	T-Value	Probability Level	Reject H0 at .050	Power (Alpha=.050)	Power (Alpha=.010)
Difference <> 0	0.9138	0.364068	No	0.146899	0.046343
Difference < 0	0.9138	0.817966	No	0.005392	0.000636
Difference > 0	0.9138	0.182034	No	0.229599	0.076270

Tests of Assumptions Section

Assumption Value	Probability	Decision(.050)
Skewness Normality (C2=1)	0.8441	0.398622
Kurtosis Normality (C2=1)	-0.9122	0.361687
Omnibus Normality (C2=1)	1.5445	0.461971
Skewness Normality (C2=2)	1.5873	0.112434
Kurtosis Normality (C2=2)	-0.6499	0.515734
Omnibus Normality (C2=2)	2.9421	0.229686
Variance-Ratio Equal-Variance Test	1.0198	0.954656
Modified-Levene Equal-Variance Test	0.5050	0.479763

Median Statistics

Variable	Count	Median	95.0% LCL of Median	95.0% UCL of Median
C2=1	35	87.5	75	87.5
C2=2	35	87.5	75	87.5

Mann-Whitney U or Wilcoxon Rank-Sum Test for Difference in Medians

Variable	Mann Whitney U	W Sum Ranks	Mean of W	Std Dev of W
C2=1	685.5	1315.5	1242.5	76.88925
C2=2	539.5	1169.5	1242.5	76.88925

Number Sets of Ties = 3, Multiplicity Factor = 63210

Alternative Hypothesis	Exact Probability		Approximation Without Correction			Approximation With Correction		
	Prob Level	Reject H0 at .050	Z-Value	Prob Level	Reject H0 at .050	Z-Value	Prob Level	Reject H0 at .050
Diff<>0			0.9494	0.342408	No	0.9429	0.345725	No
Diff<0			0.9494	0.828796	No	0.9559	0.830444	No
Diff>0			0.9494	0.171204	No	0.9429	0.172862	No

Kolmogorov-Smirnov Test For Different Distributions

Alternative Hypothesis	Dmn Criterion Value	Reject H0 if Greater Than	Test Alpha Level	Reject H0 (Test Alpha)	Prob Level
D(1)<>D(2)	0.114286	0.3251	.050	No	0.9794
D(1)<D(2)	0.000000	0.3251	.025	No	
D(1)>D(2)	0.114286	0.3251	.025	No	

