Automatic Speech Recognition

of the isiZulu Language

Submitted to the Department of Electronic and Computer Engineering

in the Faculty of Engineering and the Built Environment

at the Durban University of Technology,

in fulfilment of the academic requirements for the Degree

Master of Engineering

By

Nokwanda Shezi

20615270

ABSTRACT

A key component of artificial intelligence is human-to-machine communication. Such communication has been realised through virtual assistants such as Apple's Siri, Google's Now, Amazon's Alexa, etc. This technology is made possible through Automatic Speech Recognition (ASR). Only in recent years have the previously marginalised or developing countries started researching ASR for their indigenous languages. This research focuses on ASR for isiZulu, which is one of South Africa's most spoken indigenous language. The research involves two main fields of study i.e., digital signal processing (DSP) and machine learning (ML). DSP was applied in word boundary estimation and feature extraction. Machine learning was used to convert the word boundary estimation problem to a classification problem as well as for word recognition. Word boundary estimation achieved an accuracy of 68.4%, which is on par with the current research. The Mel-frequency cepstrum coefficient (MFCC) was used for the feature extraction of the speech and deep neural networks were chosen for the ML component. For the detection and classification of a word in a sentence, the trained neural network was tested by considering the effect of including or excluding explicit boundaries on the overall recognition. Word recognition accuracy with manually demarcated boundaries was 78.18%. In sentence recognition accuracy achieved without demarcated boundaries was 17.74% while a 23.28% accuracy was achieved where boundaries are demarcated using classification. While in-sentence recognition accuracy for the two algorithms was both low, the accurately recognised words were determined by different heuristics. Other factors, such as the complex differences between the indigenous isiZulu language and other more commonly spoken languages, are also highlighted and further research avenues are proposed.

PUBLICATIONS

A component of this research entitled “Word Boundary Estimation of isiZulu Continuous Speech” was presented at the IEEE International Conference on Emerging Trends in Engineering Science and Technology (ICETEST2020).

DECLARATION

This dissertation is the student's own work, every cited work or text have been properly referenced. It has not been partially or fully submitted at any other University.

This research was duly supervised by Dr S. Reddy at the Durban University of Technology.

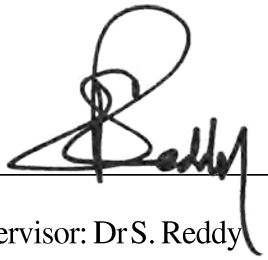
Submitted by:

15 November 2021

Nokwanda Shezi — Date

Student Number: 20615270

Approved for Final Submission by:



15/11/2021

Supervisor: Dr S. Reddy — Date

ACKNOWLEDGEMENTS

First and foremost, I would like thank God the Almighty and my ancestors for my spiritual wellbeing throughout my MEng journey. I would like to thank my parents Khawa and MaMsomi for the support, encouragement, and tough love throughout my academic career.

I would also like to thank my supervisor Seren Reddy for guidance and valuable input into the direction and technical aspect of my research, as well as the structure of my dissertation.

LIST OF ACRONYMS

4IR : 4th Industrial Revolution

AdaBoost : Adaptive Boost

Adam : Adaptive Moment Estimation

ADC : Analog to Digital Converter

ANN : Artificial Neural Network

ASR : Automatic Speech Recognition

Bagging : Bootstrap Aggression

CEP : Cepstrum Pitch Determination

CNN : Convolution Neural Network

CSIR : Council for Science and Industrial research

DAC : Digital to Analog Converter

DARPA : Defence Advanced Research Program

DFT : Discrete Fourie Transform

DL : Deep Learning

DNN : Deep Neural Network

DSP : Digital Signal Processing

ECOC : Error-Correction Code

EEF : Energy Entropy Features

FPR : False Positive Ratio

GPE : Gross Pitch Error

GRU : Gated Recurring Unit

HLT : Human Language Technology

HMM : Hidden Markov Model

HOS : Higher-Order Statistics

IBM : International Business Machine Corporation

***k*-NN** : K-Nearest Neighbour

LHS : Log-Harmonic Summation

LPC : Linear Predictive Coding

LSTM : Long Short-Term Memory

LSTM : Long Short-Term Memory

MDC : Minimum Distance Classifier

MFCC : Mel-Frequency Cepstral Coefficients

MIT : Massachusetts Institute of Technology

ML : Machine Learning

MLP : Multi-Layer Perceptron

MTL : Multitask Learning

NB : Naïve Bayes (classifier)

NCHLT : National Centre for Human Language Technology

NFC : Normalised Correlation Function

NLP : Natural Language Processing

PCM : Pulse Code Modulation

PEF : Pitch Estimation Filter

PLP : Perceptual Linear Prediction

RAC : Radio Corporation of America

RFC : Random Forest Classifier

RMS : Root Mean Square

RMSProp : Root Mean Square Propagation

RNN : Recurring Neural Network

ROC : Receiver Operating Characteristic

ROS : Random Over-Sampling

RUS : Random Under-Sampling

RUSboost : Random Under-Sampling with Boost

SA : South Africa

SC : Spectral Centroids

SGD : Standard Gradient Descent

SGDM : Stochastic Gradient Descent with Momentum

SNR : Signal to Noise Ratio

SRH : Summation Residual Harmonics

STE : Short-Term Energy

STT : Speech-To-Text

SVM : Support Vector Machine

TI : Texas Instrument Inc

TPR : True Positive ratio

TTS : Text to Speech

VAD : Voice Activity Detector

VAT : Voice Activated Typewriter

WAV : Waveform Audio Format

WCC : Wavelet Cepstral Coefficients

WER : Word Error Rate

ZLC : Zero Line Crossing

LIST OF NOTATIONS

$S(t)$	Resultant signal
$s(t)$	Speech signal
$n(t)$	Noise signal
L	Length
β	Gradient decay factor
θ	Parameter vector
α	Learning rate
h_v	Vocal tract impulse
f	Frequency
ϵ_t	Pseudo - loss
$E(.)$	Expectation operator
μ	Mean
σ	Standard deviation
Kurt	Kurtosis
$\widetilde{\mu_3}$	Skewness
$R(\tau)$	Pitch candidate
λ	Variance
ε	Priori SNR
γ	Posteriori SNR

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

INTRODUCTION

1.1 Background

The fourth industrial revolution (4IR) will have a significant impact on developing countries such as South Africa. Previous industrial revolutions marginalised vulnerable societies. However, with 4IR and the impact of globalisation, these societies must be shaped to ensure inclusivity and fairness. The main characteristic of 4IR is artificial intelligence and the digital disrupting technologies brought about by the merging of advanced hardware and software; this will affect all industrial and economic sectors and have a significant effect on the entire society.

The ability of machines to mimic human behaviour is a great subject of interest in the engineering field, particularly the capability to communicate in natural language and respond to spoken language. Automatic Speech Recognition (ASR) is a significant part of machine-to-human communication. In recent years ASR has equipped mobile devices and homes with virtual assistants such as, among others, Apple's Siri, Google's Now, and Amazon's Alexa.

Most of the developed countries began researching ASR, using their respective languages, as early as the 1930s. Making machines "understand" indigenous languages is an important step in ensuring that there is inclusiveness of developing nations in 4IR.

1.1.1 Overview of an ASR system

ASR converts a sequence of spoken words into their textual representation. The algorithm for ASR depends on the type of natural speech being processed. Five types of natural speech were identified [1]; these are shown in Table 1.1.

Table 0.1: Types of Natural Speech

Speech Type	Identification
Spelled speech	Each word is spelled out with a pause between letters or phones.
Isolated speech	Words are uttered individually with a pause at the end of each word.
Continuous speech	The speaker does not have any pauses between words e.g. a formal speech.
Spontaneous speech	A dialogue type of speech where one human is communicating with another human.
Highly conversational speech	A discussion between several people e.g., meeting.

ASR algorithm also depends on the size of the vocabulary or lexicon. Whittaker [2] classified the lexicon sizes as small (0 to 1 000 words), medium (1 001 to 10 000 words), large (10 001 to 100 000 words), and very/extra-large (>100 000). The size of the vocabulary is important as most ASR systems use statistical models that attempt to model all the words of a language.

Another factor to consider in ASR algorithm development is language resources. Western languages have a myriad of resources while most indigenous languages are under-resourced for ASR purposes. The term "under-resourced languages" is described by Krauwer [3] as languages that (i) lack a unique writing system or stable orthography, (ii) have limited presence on the web, (iii) lack linguistic expertise, and (iv) have a lack of electronic resources for speech and language processing, such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, and vocabulary lists.

The state-of-the-art ASR system generally aims to determines the most probable word sequence of orthographic word $\hat{W}$, from a number of possible sequences W, given observed

acoustic features O. This concept is illustrated in fig 1-1 for a statistical speech recognition system.

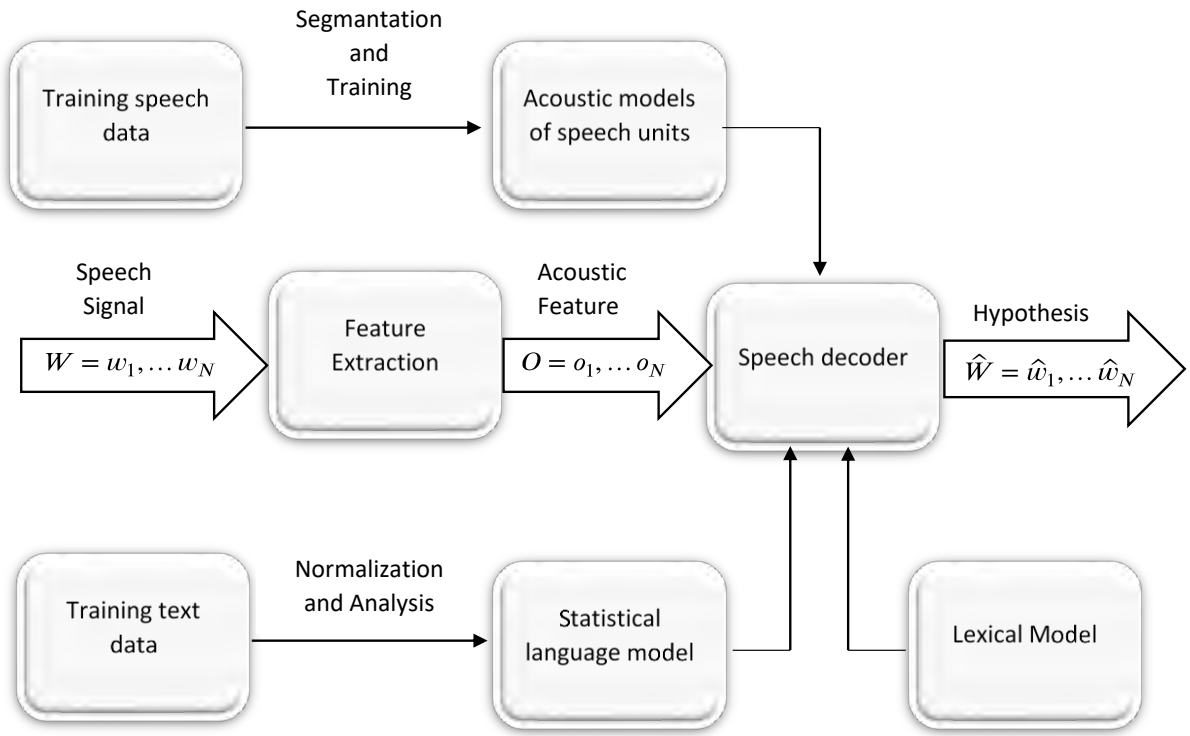


Figure 0-1: Architecture of a state-of-the-art ASR System and its Components[1]

The speech signal feature vector (O) is extracted from the acoustical signal. The acoustic model of the speech decoder matches the feature vector to a word sequence with similar features. The language model determines the likelihood of each hypothesis. Thus, the speech decoder can be represented by determining the word sequence $\hat{W}$, that maximises the likelihood of the sequence given feature vector O.

$$\hat{W} = arg \max_{W}(W|O) \tag{1.1}$$

Using Baye's rule to determine the conditional probability of word sequence W given feature vector O gives:

$$P(W|O) = P(W)\frac{P(O|W)}{P(O)} \tag{1.2}$$

$P(O)$ is independent of W,

$$\therefore\ \hat{W} = arg\max_{W} P(W)P(O|W) \tag{1.3}$$

It can then be concluded that the acoustic model determines the conditional probability of sequence W given feature vector O; $P(O|W)$. The language model determines the unconditional probability of word sequence W. These two models operate at word recognition level. A lexical model is used to determine the phonetic transcript of the words. Thus, assigning the correct spelling to the words in the sequence.

1.1.2 Automatic Speech Recognition History

ASR has been a subject of interest since the 1930s when Homer Dudley of Bell Laboratories proposed a system model for speech analysis and synthesis [4]. The interest significantly grew in the research community and the general population upon the introduction of the first mobile virtual assistant, Siri, in 2011 [5]. Moreover, ASR has become the driving force behind various machine learning (ML) algorithms hence the increasing mutual influence between the two communities [6].

In 1888 Alexander Graham Bell and Charles Sumner Tainter formed Volta Graphophone Co. to manufacture machines for recording and reproducing sound. After that, machines were invented to help secretaries with dictations, followed by the electronic typewriter that could take dictation. A speaker-dependent voice-activated typewriter was later developed by a team

at International Business Machines Corporation (IBM) led by Fred Jelinek [7]. Bell Laboratories also built a system for isolated digit recognition for a single speaker in 1952 [8].

In the early development of speech recognition, the concept was based on theoretical acoustic phonetics, which describes the phonetic element. Computers were not fast enough to process sophisticated programs; hence most developments were hardware-based. In 1956 Olson and Belar of Radio Corporation of America (RCA) Laboratories developed the first 10-word mechanical recogniser [9]. In 1959 the University College in England improved on this recogniser by designing an automated phoneme recogniser for four vowels and nine consonants [10]. The system incorporated statistical information about allowable phoneme sequences in English, which increased the overall phoneme recognition. In the same year, Forgie and Forgie at the Massachusetts Institute of Technology (MIT) Lincoln Labs built a speaker-independent 10-vowel recogniser [8]. Japan contributed to numerous advancements, the most memorable being the continuous speech recogniser developed by Kyoto University in 1962 using a hardware segmenting system and zero-crossing analyser [11]. In 1966 Reddy at Carnegie Mellon University conducted a ground-breaking continuous speech recognition research [12]. Following this research, there were rapid developments in speech recognition achieving significant milestones in each decade.

In the 1970s, Itakura showed how linear predictive coding (LPC) spectral parameters can be used in speech recognition [13]. Subsequently, in 1979, Itakura collaborated with Rabiner et al. in developing a speaker-independent speech recognition system using the k-nearest neighbour rule [14]. In the 1980s speech recognition point of interest shifted to recognising a string of connected words, which called for a methodology shift from a template based approach to a statistical modelling framework. The most significant technology developed in this era was the hidden Markov model (HMM).

In the 1990s, pattern recognition was transformed into an optimisation problem involving the minimisation of recognition error. This concept produced discriminative training and kernel-based methods [15]. Current studies focus on the deep learning approach, emotion recognition, and development of high-performance ASR systems.

1.1.3 Under-Resourced Languages and Language Extinction

It is estimated that in the next century, around half of currently existing languages will be extinct; on average, one language dies every two weeks [1]. Of the 7 117 spoken languages today, 40% are in danger, and only 10 languages account for more than 65% of the global population [16]. Languages can only be saved if its community wants it and the surrounding culture respects this wish. Including the languages in technological developments is a big part of saving languages. The world's top 10 languages shown in Table 1.2, have had their ASR systems in development since the first half of the 20th century. In contrast, some of the languages in the emerging nations have only been in development from the early years of the 21st century. Including indigenous languages in this new technological progression may aid in averting language extinction.

Table 0.2:Global top 20 languages [16]

Rank	Language	Speakers (million)
1	English	1268
2	Mandarin Chinese	1120
3	Hindi	637
4	Spanish	538

5	French	277
6	Standard Arabic	274
7	Bengali	265
8	Russian	258
9	Portuguese	252
10	Indonesian	199

The biggest challenge for under-resourced languages is that porting human language technology (HLT) systems goes beyond just retraining existing models. Processing new languages present additional challenges such as special phonological systems, word segmentation problems, fuzzy grammatical structures, and unwritten languages.

Most African languages are considered under-resourced as per the definition. As a means of preserving these languages, the South African Department of Arts and Culture appointed the Meraka Institute at the Council for Scientific and Industrial Research (CSIR) to develop speech resources for all 11 official languages in South Africa. These resources were archived and made available as an open-source database at the National Centre for Human Language Technology (NCHLT) [17].

IsiZulu is one of South Africa's 11 official languages and mother tongue to some 11.6 million South Africans, which is around 22.7% of the population [17]. Although isiZulu is one of the most widely spoken indigenous languages in South Africa, it is nevertheless considered to be an under-resourced language.

1.2 Study Objectives

This study aims to investigate isiZulu language acoustic features and possible classification algorithms for automatic continuous speech recognition, focusing on acoustic modelling. The study could contribute to the improvement of human-machine communication in South Africa and beyond as well as play a role in the decolonisation of education.

The NCHLT has established a substantial corpus for South Africa's under-resourced languages. This research focuses on using data from this corpus to investigate word recognition using four types of machine learning algorithms; these include recurrent neural networks (RNN), RNN and long short-term memory (RNN-LSTM), random under-sampling with adaptive boost (RUSboost), and RNN and LSTM (RUSboost-RNN-LSTM).

The research has four main components:

i. Identify the most useful signal information in isiZulu speech using Digital Signal Processing (DSP).
ii. Convert the speech signal to text using machine learning algorithms.
iii. Compare training time and accuracy of the different ML algorithms.
iv. Compare the accuracy of the ASR model to other similar techniques.

1.3 Methodology Overview

DSP is the standard for audio and speech processing. Audio signal processing converts raw analogue input signals into a digital format that can be mathematically analysed. DSP block system for speech recognition has three variations; these are (i) a complete audio system, (ii) an audio recognition system, or (iii) an audio synthesis system [18]. These are illustrated in Figs. 1-2, 1-3, and 1-4, respectively. Each variation is made up of a combination of amplifiers,

analogue to digital converters (ADCs), processing blocks, and digital to analogue converters (DACs).

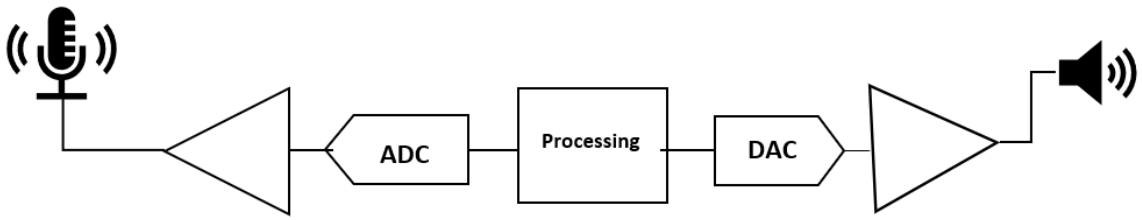


Figure 0-2: Complete Digital Audio Processing System

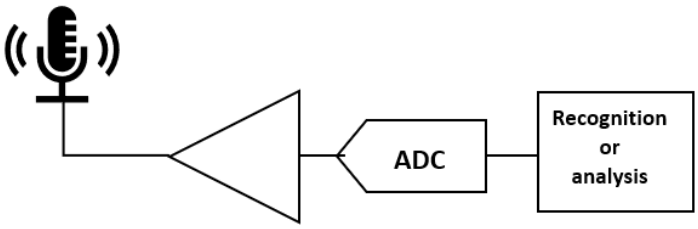


Figure 0-3: Audio/Speech Recognition System

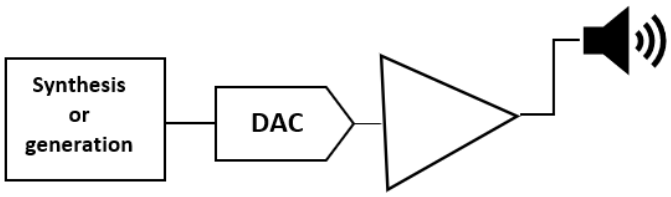


Figure 0-4: Speech/Audio Synthesis System

This study is primarily concerned with audio/speech recognition, which is illustrated in Fig. 1-3. This system comprises amplifiers, an ADC, and a recognition/analysis block. The aforementioned NCHLT corpus [17]contains audio files formatted using single-channel

waveform audio format (WAV), with 16-bit signed integer pulse code modulation (PCM) encoding and at a sample rate of 16kHz. These WAV files are used for feature extraction and classification.

This study investigates the acoustic modelling of the isiZulu language using the NCHLT corpus. The acoustic model determines the conditional probability of a word sequence given an acoustic feature vector. One way of achieving this is through discriminating classes of context-independent speech units based on their acoustic features. Thus, representing audio signal by classes, this process is called classification.

The state-of-the-art ASR system operates in two modes: model training and speech decoding. As part of the ASR system, acoustic modelling has two phases: training and testing, as illustrated in Fig 1-5, which is also the basis of this study.

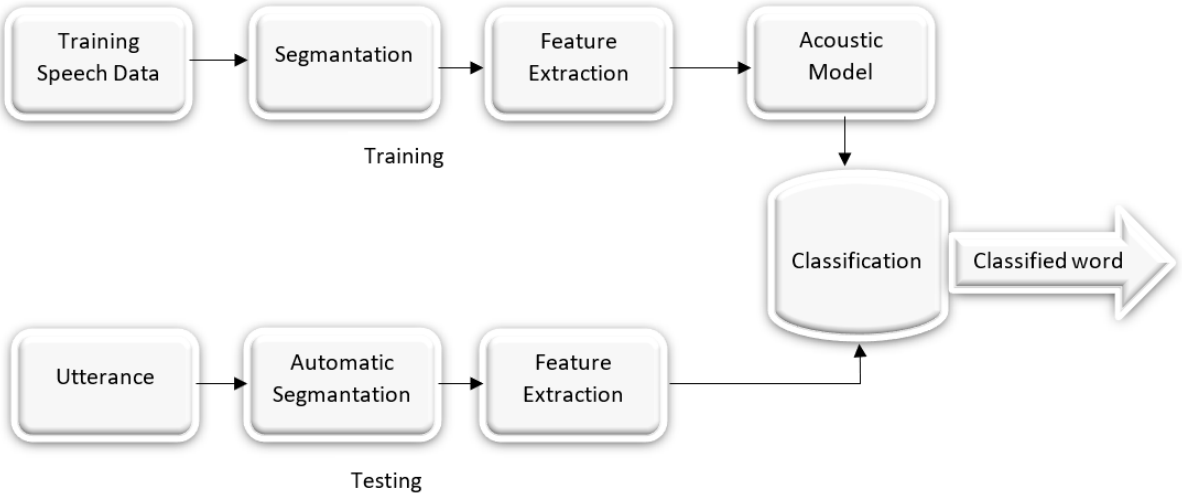


Figure 0-5: Architecture of Acoustic Modelling System and its Components

There are two approaches to word classification; the first method attempts to match the acoustic features to a word with similar features by processing the entire sentence, while the second

method involves the initial segmentation of the sentence into individual words, followed by matching the acoustic features to a word with similar features. The classifiers were trained using a set of ten words and the two algorithms were compared based on accuracy using the confusion matrix. The accuracy was calculated as the percentage of all correct predictions from all predictions.

For the latter approach, sentence segmentation was achieved by analysing acoustic and higher-order statistic (HOS) features. The segmentation was made into a classification problem by applying an ensemble classifier. Word features were extracted using Mel-frequency cepstral coefficients (MFCC) and word classification was achieved using a Deep Neural Network (DNN).

All results were conducted using MATLAB 2020a on a computer containing an Intel® Core™ i7-8550 CPU @ 1.8 GHz, 8 GB RAM, and running a Windows 10 64-bit operating system.

1.4 Scope Limitation

This research is based on the recognition of only a limited number of isiZulu words. A full ASR language model is outside the scope of this research.

1.5 Structure of the Dissertation

Chapter 2 proposes the use of prosodic features and two HOS to investigate the heuristics associated with the boundaries of isiZulu words. These features were used to train a boundary recognition algorithm.

Chapter 3 proposes MFCC for the feature extraction algorithm. Six steps of this algorithm are discussed. To determine the number of coefficients with the highest accuracy, a k-nearest neighbour classifier was used.

Chapter 4 proposes a word recognition algorithm using isolated words. Ten words were chosen from the NCHLT database. This database has limited samples, therefore supervised learning was proposed. The two types of supervised learning; namely classification and regression are discussed. The former was selected. Three popular algorithms are also discussed, and DNN is selected based on current trends.

Chapter 5 proposes two other methods of word classification; the first method attempts word correlation by processing the entire sentence, while the second method involves the initial segmentation of the sentence into individual words, followed by word correlation. The two approaches are then compared based on classification accuracy.

Chapter 6 provides a conclusion of the dissertation.

DATA COLLECTION AND PREPARATION

1.6 Introduction

The Meraka Institute at the CSIR was appointed by the South African Department of Arts and Culture to develop speech resources for all 11 official languages in South Africa. As of 2013, the resources were archived and made available at the NCHLT [1]. The study aimed to create an open-source corpus for, among others, ASR and Text-to-Speech (TTS) conversion development. For each language, data was collected such that there was an even distribution of participants across rural, urban, gender, and age groups. The final corpus for isiZulu included 25 650 vocabulary words from 210 speakers, 130 866 tokens with a total duration of 56 hours, and 14 minutes [17]. The corpus primarily consists of sentences comprising an average of 4 words.

In the state-of-the-art ASR system architecture, acoustic modelling requires isolated basic speech units such as monophones, syllables, allophones, triphones, and pentaphones. Language modelling requires all possible sequences of the basic units. On application, the acoustic features of the utterance are extracted without delineation of basic unit boundaries. The features are then matched with the most likely basic unit. Then the language model determines the most likely sequence. This process works well for well-resourced languages, as language modelling requires vast amounts of data. For most under-resourced languages, language modelling is a challenge. It becomes a more significant challenge where continuous speech recognition is required.

The abovementioned challenges conclude that importing an under-resourced language into an existing system is more complex than re-training the model. One way to address this challenge

is by implementing an algorithm that will delineate word boundaries in a continuous speech before extracting acoustic features. Recognition of the beginning and end of words is a critical task in speech recognition [19]. Implementing a word boundary algorithm before word recognition has several benefits in ASR, including reduced processing in the intended use of the speech as well as increased efficiency in recognition of out of vocabulary words, such as proper nouns [20].

One algorithm developed to detect the start and end of a word is the Voice Activity Detector (VAD). The VAD was developed for the English language based on the signal-to-noise ratio (SNR) and assumes that there is a short pause between words [21]. In isiZulu, words end in vowels and mostly begin with vowels. Vowels are louder compared to consonants; hence the end of isiZulu words generally have higher energy. Moreover, isiZulu has a few phonemes not found in English, such as the clicks (alveo- lateral, dental, and palatal) and some affricates including but not limited to Eφv, [tʃ ´], [nts´], [ndz], [ntɓ´], [ntɓµ´], [ndɧ], [kɓ ʼ]and [ŋkɓ ʼ]. Some of these phonemes are voiceless such as [kɓ´], [tʃ ´], and ntɓ´; this may compromise the accuracy of the VAD due to that the VAD analyses a frame based on the prior frame's noise. An experiment to prove this hypothesis was conducted using the MATLAB Audio toolbox. Voice activity was checked per frame of 25 ms and a 75% frame overlap. The results are illustrated in Fig 2-1, where the value 1 in the VAD plot represents a 100% probability of voice activity. The audio recording has 3 words. However, the VAD estimates that the recording has 5 words. Consequently, the VAD model on its own becomes relatively ineffective for isiZulu word boundary detection.

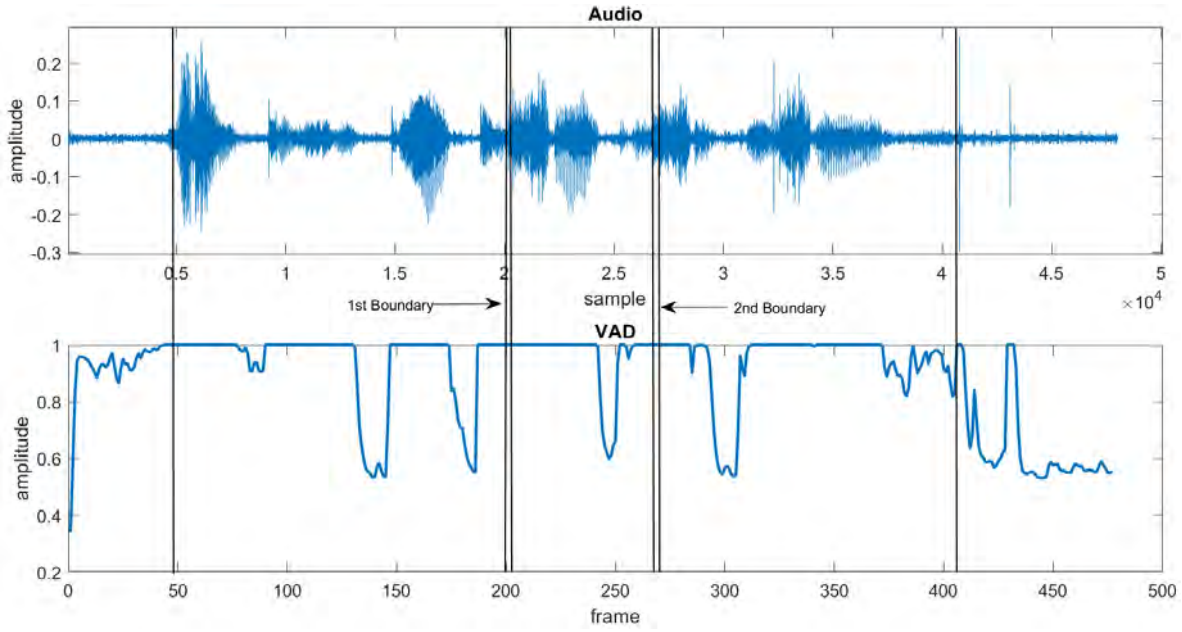


Figure 0-1: VAD for "Abantu Abafuna Isondlo"

This chapter proposes the use of prosodic and HOS features for word boundary estimation as well as a supervised learning classifier for automatic boundary detection of isiZulu speech. The prosodic features are extrapolated using the discrete Fourier transform (DFT), log energy, zero-line crossing (ZLC), pitch frequency, and frame location. For the HOS features, both skewness and kurtosis are considered. For the classification of the speech segments, a tree ensemble classifier is proposed.

Section 2.2 discusses the physiological production of speech. Section 2.3 presents related word boundary estimation techniques. Section 2.4 discusses the proposed approach. Results are presented and discussed in section 2.5. The chapter is concluded in section 2.6.

1.7 Speech Signal Production

The human voice is produced primarily through a combination of air movements in and out of the lungs and glottis that generates vibrations into the vocal tract and is filtered through the surrounding muscles[22]; this is illustrated in Fig. 2-2.

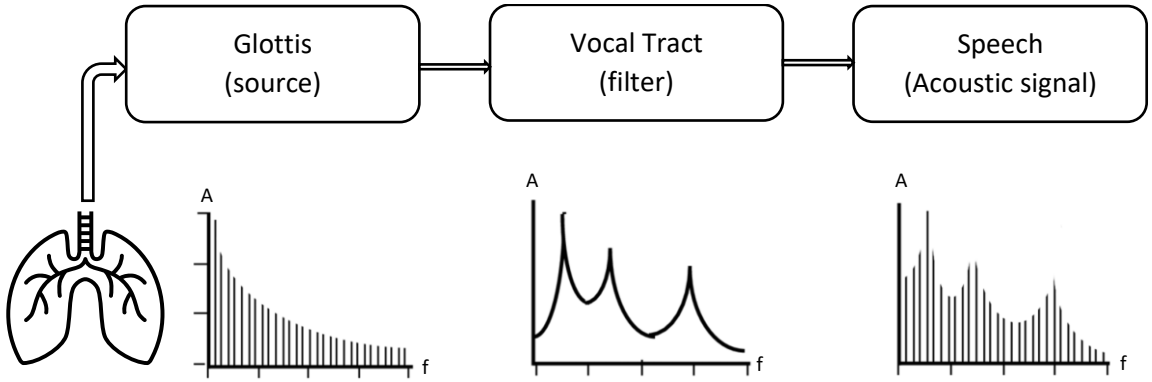


Figure 0-2: Source-Filter Speech Production Block Diagram

According to physiology, speech is produced by the lungs, trachea, larynx, pharyngeal cavity (throat), oral cavity (mouth), and nasal cavity[23]; this is illustrated in Fig. 2-3. The buzzing sound of vocal vibration has no articulation but pitch frequency information. This sound is due to the uniformed pulses generated by the glottis. The area above the trachea, called the vocal tract, is the lip, jaws, tongue, and velum responsible for speech articulation. The study of this speech production component assists in determining the formation of the acoustic signal, thus facilitating a more informed signal analysis. The signal analysis would then inform the digitization and processing of the speech, facilitating the development of algorithms for speech recognition (speech-to-text) and speech synthesis (text-to-speech). Tone and pitch information assists in detecting voice activity and emotion recognition.

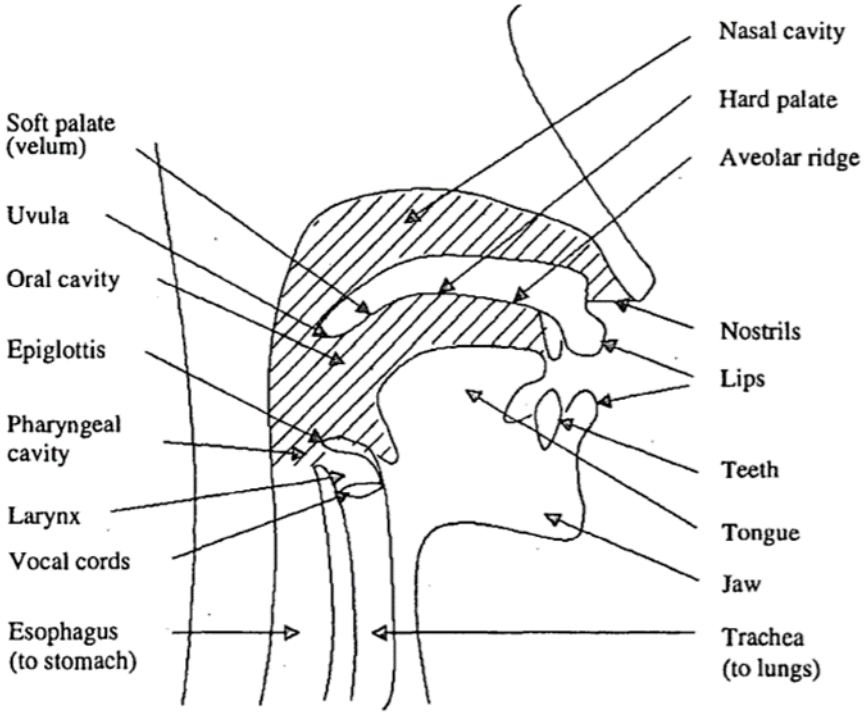


Figure 0-3: Physiological Illustration of the Human Speech Production [23]

1.8 Related Work

There are two broad methods proposed for end-point detection; these are referred to as explicit and implicit. The former is a stand-alone process, while the latter is inclusive in the recogniser. To be able to control the endpoint detection sensitivity without sacrificing recognition accuracy, an explicit algorithm is favourable [2]. Boundary estimation may be considered as a binary sequence classification scheme since the fundamental aim is to determine if there is either voice activity, which is represented by an output of 1, or no voice activity, which is represented by an output of 0 [5].

Agarwal et al. proposed using utterance intensity and pitch frequency for boundary estimation for the Hindi language [24] . Their database was self-collected, consisting of 3 speakers, with

each recording 40 sentences in 5 different emotions, totalling 600 recordings. The aim was to estimate the boundary regions, thus estimating the word count. Overall results were based on 3 emotions, namely neutral, happy, and angry. The average accuracy was 88%, 91%, and 79%, respectively.

In addition to utterance intensity and pitch frequency, Naganoor et al. [25] proposed ZLC, log energy, and two HOS features (skewness and kurtosis) to detect the endpoint. For their research, they used an American English database recorded by Texas Instrument Inc (TI) and transcribed by the (MIT). The database was named after the two institutes (TIMIT). The database consists of 630 speakers of which 438 are male and 138 are female [26]. Each speaker recorded 10 sentences therefore making a total of 6300 recordings. Supervised learner classifiers were used to make the boundary estimation a classification problem. In this approach, they considered the effects of noise and the co-articulation between the end of the previous word and the beginning of the current word. They used a frame width of 20 ms with a 50% overlap. A predicted boundary was considered correct if it was within 10 frames of the actual boundary. The addition of HOS improved the boundary estimation by 46.70% compared to only considering rudimentary features. For classification, the use of an ensemble classifier showed the best results, compared to using a support vector machine (SVM), a random forest classifier (RFC), or an artificial neural network (ANN). Another study considered a combination of short-time energy (STE) and intensity when analysing a 1-hour news broadcast [27]; this achieved an accuracy of 67.15%. Patil and Pardeshi used STE and spectral centroids on a smaller self-collected dataset, with 800 sentences from 4 speakers, where an accuracy of up to 97% was achieved [28].

1.9 Sentence Segmentation Database

For this research, five sentences were extracted from the NCHLT database; the selected sentences are listed in Table 2.1. Recordings were split into training data and testing data in a 60:40 ratio, respectively. The database selection was made so that there will be enough training data containing words with enough samples for training ASR algorithms. To analyze the impact of different sounds, the database had to have a broad range of syllables. Considering the database requirements, the selection criteria for the sentences were as follows.

- Sentences had to be repeated more than 90 times in the database;
- At least one word in the sentence must be repeated at least 200 times by different users;
- Syllable diversity; and
- Each sentence must be repeated at least 3 times by each speaker.

Table 0.1: Word Boundary Estimation Database

Sentence number	Sentence	Number of speakers	Male	Female	Repetition per speaker	Total recordings	Training samples	Testing
1	“ehhovisi eleseduze labezindaba”	33	15	18	3	**99**	**59**	**38**
2	“iphepha lokuqala p”	32	17	15	3	**96**	**58**	**38**
3	“intela ngendlela evikelekile”	31	15	16	3	**93**	**56**	**37**
4	“abantu abafuna isondlo”	41	22	19	3	**123**	**66**	**44**
5	“isitifikedi sakho sokuzalwa”	33	15	18	3	**99**	**59**	**40**

The audio files were sampled at 16 kHz. For speech production, it takes approximately 20 ms to 30 ms for a vocal tract to change its size and shape [11]; this is the standard framing range in ASR [24, 25, 29]. In the proposed approach, for all feature extraction, the sampled signal was segmented in windows of 20 ms with an overlap of 50%. Recordings were manually labelled into three categories; these include noise, boundary, and token. The labeling was done using MATLAB signal processing and communication toolbox, as illustrated in figure 2.4 below. Thereafter, word demarcations were tabulated. All segments were defined as a group of frames. Only boundaries occurring between words were considered. During the manual sentence segmentation, the boundary regions were measured at an average of 94 ms, i.e., approximately 10 frames. Boundary regions were then determined as a group of frames. Boundaries were demarcated at the first syllable of the words. Therefore 10 consecutive frames before the demarcated boundary were labelled as boundary frames. The final frame labels were 44 391 noise frames, 93 200 voice frames, and 12 814 boundary frames. Noise frames indicate a pause or stop depending on the number of noise frames.

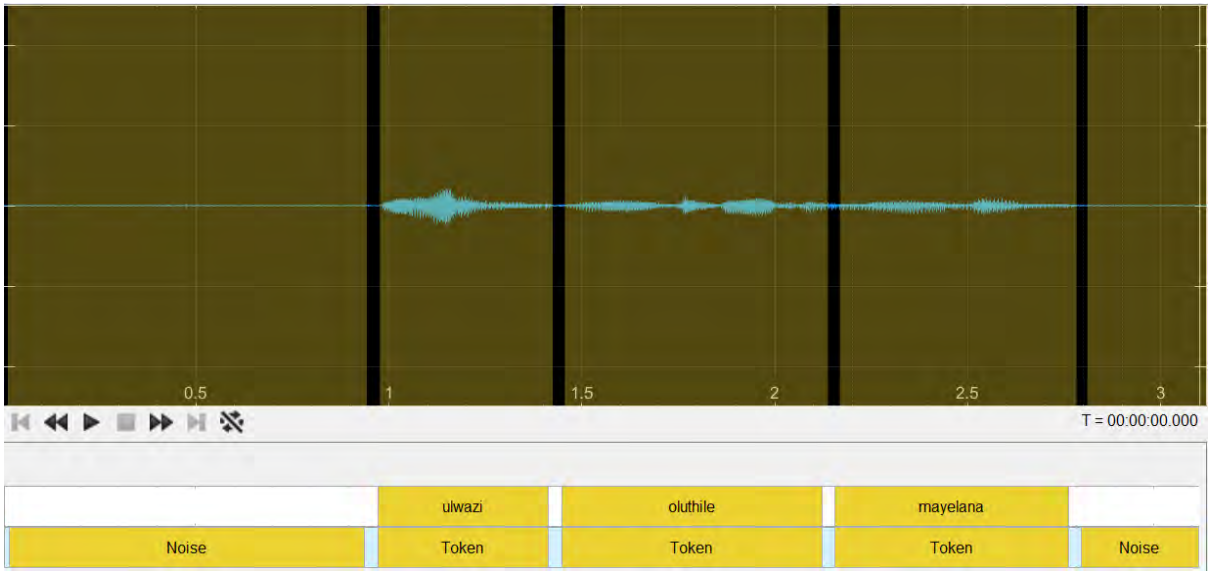


Figure 0-4: Audio Labelling Using MATLAB Signal Processing and Communication

1.10 Word Boundary Features

The voice component of speech is assumed to have a higher amplitude compared to the non-voice and noisy segments; this assumption is highly probable for natural/physiological speech processing. This is illustrated in Fig. 2-5 and mathematically described by Eq. (2.1). Another assumption is that background noise remains stationary for long periods; however, for machine processing, this assumption does not necessarily hold, as microphone sound effects, electrical noise, and the non-voiced syllables must be accounted for. Consequently, to increase the word boundary detection accuracy, four rudimentary features were initially considered; these include Gaussian statistic modelling of the DFT, log energy, ZLC, and pitch frequency. Moreover, two additional HOS features were also considered viz. skewness and kurtosis. For this proposed approach the baseline experiments were conducted using a single utterance "ulwazi oluthile mayelana".

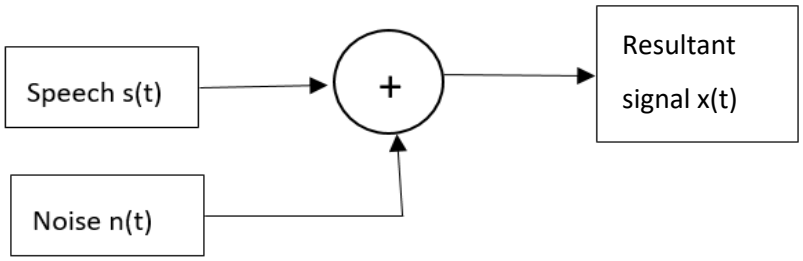


Figure 0-5: Resultant Signal Block Diagram

$$s(t) + n(t) > n(t) \tag{2.1}$$

1.10.1 Discrete Fourier Transform

MATLAB's audio toolbox was used to implement the DFT and Gaussian statistic model for determining the probability of voice activity in each frame of 25 ms. This model assumes that an independent additive noise deteriorates the signal of a frame; this can be represented by the two hypotheses, $\mathbf{H_0}$ and $\mathbf{H_1}$, where:

$\mathbf{H_0}$ denotes the absence of speech and $\mathbf{H_1}$ denotes the presence of speech.

$$\mathbf{H_0} \qquad X = N$$

$$\mathbf{H_1} \qquad X = N + S$$

X is determined by the probability functions given by equations (2.3) and (2.4):

$$p(X|H_0) = \prod_{k=0}^{L-1} \frac{1}{\pi \lambda_N(k)} \, exp\left\{ -\frac{|X_k|^2}{\lambda_N(k)} \right\} \tag{2.3}$$

$$p(X|H_1) = \prod_{k=0}^{L-1} \frac{1}{\pi \left[\lambda_N(k) + \lambda_S(k)\right]} \, exp\left\{ -\frac{|X_k|^2}{\lambda_N(k) + \lambda_S(k)} \right\}, \tag{2.4}$$

where N, S and X are L dimensional DFT coefficients vectors of noise n(t), voice signal s(t), and the resultant signal x(t), respectively; with $\lambda_N(k)$ and $\lambda_S(k)$ denoting the N_k *and* S_k variance, respectively. The functions are based on the Gaussian statistical model that the DFT coefficients of each process are asymptotically independent Gaussian random variables [21].

The likelihood of the k^{th} frequency band is represented by Eq. (2.5) below:

$$\Delta_k \triangleq \frac{p(S|\boldsymbol{H_1})}{p(S|\boldsymbol{H_0})} = \frac{1}{1+\ \varepsilon_k}\ exp\left\{\frac{\gamma_k \varepsilon_k}{1+\varepsilon_k}\right\} \tag{2.5}$$

where ε_k is the a priori SNR given by Eq. (2.6) and γ_k is posteriori SNR given by Eq. (2.7):

$$\varepsilon_k \triangleq \frac{\lambda_s(k)}{\lambda_n(k)} \tag{2.6}$$

$$\gamma_k \triangleq \frac{|S_k|^2}{\lambda_n(k)} \tag{2.7}$$

The implementation of the above algorithm on the baseline utterance “ulwazi oluthile mayelana” yielded an array of voice activity probability where the value 1 represents 100% probability; this is illustrated in Fig. 2-6. The experiment was conducted in MATLAB using a pre-recorded utterance from the NCHLT data base.

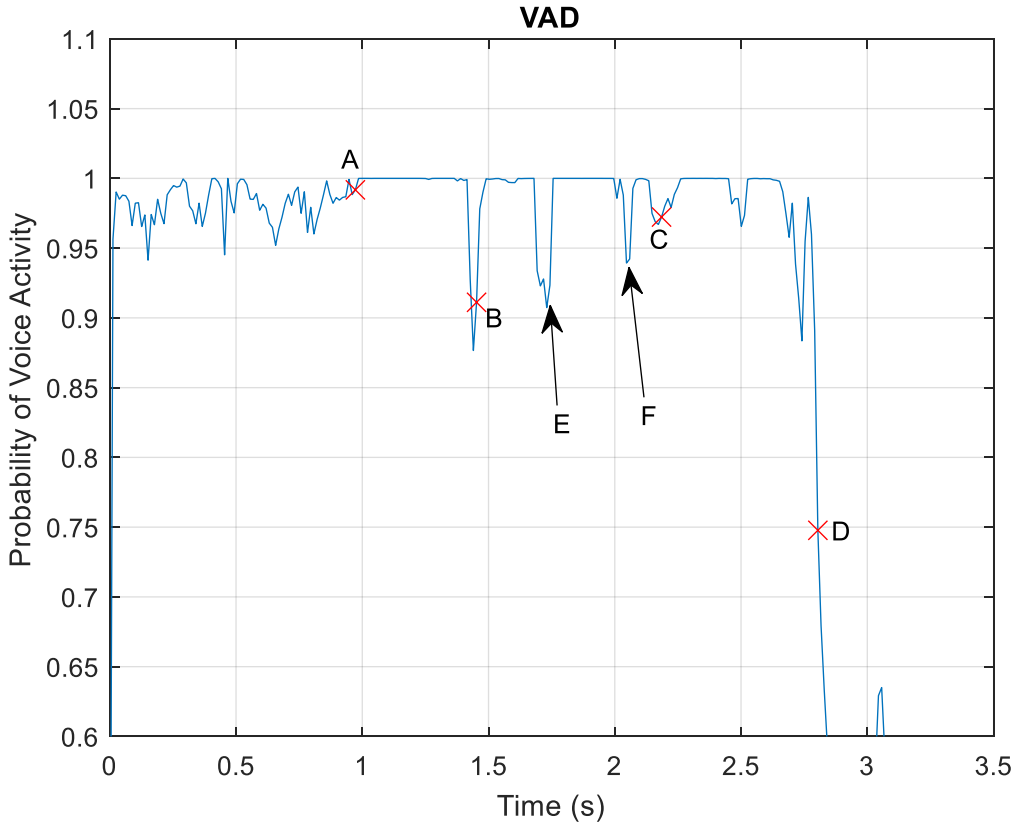


Figure 0-6:Voice Activity Probability Using DFT

For this study, a frame was considered a voiced if the probability equals or is greater than 95%. As per the manual segmentation of the utterance, point A is the beginning of the utterance, points B and C are word boundaries between the first and the second word, and the second and the third, respectively. Point A has a 96% probability of voice activity, and the subsequent frames have a 100 % probability until point B, the first-word boundary. The probability at point B is approximately 91%; after that, it increases sharply, indicating the start of the second word. Within the second word, points E and F have a probability lower than 95%. The low probability at point E is due to the unvoiced alveolar phoneme [th], with a duration of 111 ms equivalent to 9 frames. This phone is followed by a high-energy vowel [i], followed by a lateral liquid alveolar [l], then the vowel [ɛ], hence the slight drop of voice activity probability. Albeit the word boundary at Point E, the voice activity probability is above 95% due to the coarticulation

of the two words. The end of the second word ‘ε’ extends to the first phone of the third word [m], a bilabial voice.

1.10.2 Pitch frequency

The voice pitch, denoted by a lower frequency range between 50 Hz and 600 Hz, envelops the speech signal. Estimating the presence of pitch frequency assists in determining whether the frame has voice activity; silent frames consist of a higher frequency range alone. The pitch frequency was implemented using the MATLAB audio toolbox. There are five-pitch estimation methods available in the toolbox i.e. normalised correlation function (NCF)[30], pitch estimation filter (PEF)[31], cepstrum pitch determination (CEP)[32], log-harmonic summation (LHS)[33] and summation of residual harmonics (SRH) [33]. Experiments conducted by MathWorks [34] using 1 second of clean data and a 20 dB white noise factor gave results as tabulated in Table 2.2. One of the factors considered was the gross pitch error (GPE) which is measured as the percentage of pitch period detection error of more than 1 ms. This boundary detection algorithm will be implemented in an ASR system as introduced in section 2.1. Processing time is crucial for ASR applications such as real time speech-to-text. Therefore, computational cost was also considered which was measured as the time taken to process 1 s of data.

Table 0.2:Comparison of Pitch Estimation Methods

METHOD	GPE	COMPUTATIONAL COST
NCF	3.4 %	0.053 (s)
PEF	1.5 %	0.183 (s)

CEP	37.5 %	0.044 (s)
LHS	10.8 %	0.083 (s)
SRH	3.4 %	0.199 (s)

CEP had the minimum computational cost, but it also had the highest GPE. PEF had the lowest GPE, but its computational cost was too high. The method with the best trade-off between GPE and computational cost under test conditions was NCF.

The short-term NCF pitch candidate $R(\tau)$, with a speech frame length of L and an overlap length of τ was calculated using Eq. (2.8)

$$R(\tau) = \frac{1}{\overline{e_0 e_j}} \sum_{t=0}^{L-\tau} x(t)x(t+\tau)\,, \tag{2.8}$$

where x(t) is the noisy input signal as expressed in Eq. (2.1).

The resultant NCF pitch candidates are illustrated in Fig. 2-6. The recording used for this simulation was of a 22-year-old male, hence the low fundamental frequency for a significant number of frames. Point A and point D are the start and end of the utterance, respectively. Between point A and point D, the frequency remains below 300 Hz and increases sharply at point D. This feature is not affected by word boundaries; hence, it is used as a que for the start and end of an utterance.

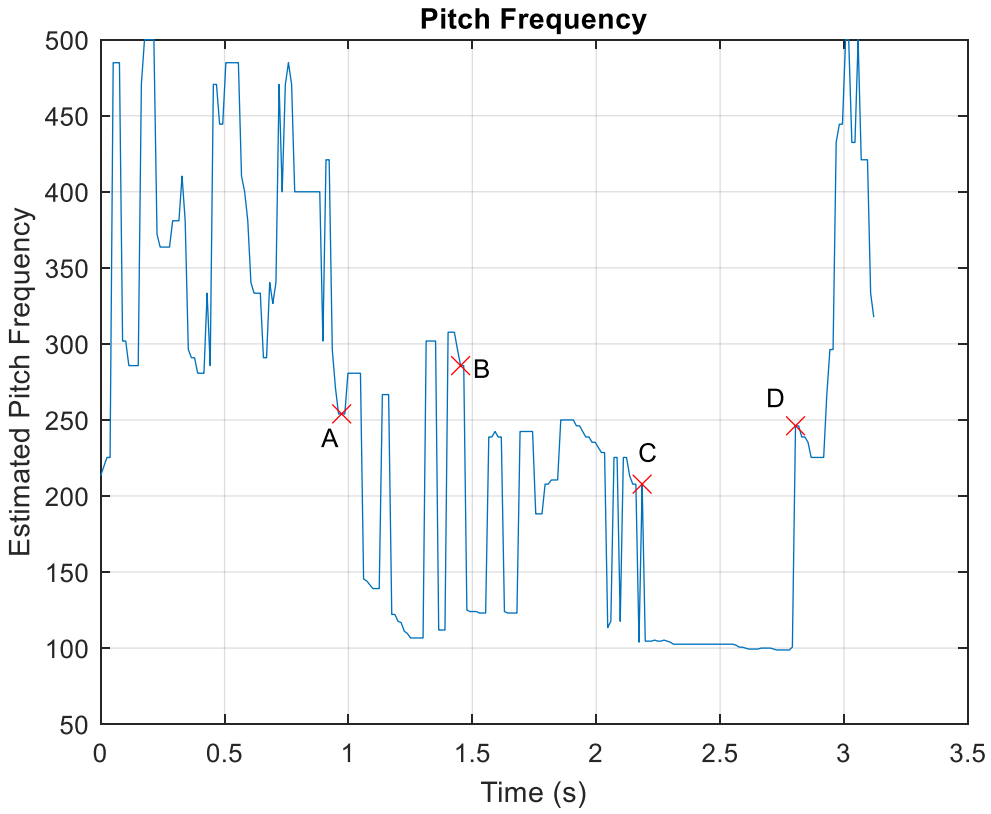


Figure 0-7: Baseline Utterance Pitch Frequency Estimation

For pitch estimation, by correlation, a frame should have at least two pitch cycles. Hence, in the proposed model a 20 ms window length, 50% window overlap, and hamming windowing were used, which resulted in 10 ms frames. Pitch frequencies were filtered using a 5^{th}-order elliptic bandpass filter with bandpass 50 Hz to 600 Hz. The stopbands (30Hz and 2000 Hz) were attenuated at 40dB with a pass-band ripple of 1 dB. A normalised correlation function (NCF) [30] was used for pitch estimation.

1.10.3 Log energy

The log energy was derived from the root-mean of each frame's energy. Log energy is low in the noisy segments and rises sharply at the beginning of a voice segment; this can be used as a cue for voice transitions and to identify the beginning of the sentence. However, in the word-to-word boundary, there is no significant change in the log energy. This is owing to the co-articulation of words, which is common in isiZulu. At the end of the sentence, the log energy falls sharply because of the short pause between sentences. The resultant log energy for frames of the baseline utterance is illustrated in Fig 2-8. Point A is the start of the utterance, point B and C are boundaries between words, the utterance ends at point D. As observed in the VAD results, points E, F and C were affected by the unvoiced alveolar phone [th], high energy vowel [i] followed by the lateral liquid alveolar [l] and the extension of the vowel [ɛ] to the first phone of the third word [m] which is a voiced bilabial. Apart from the impact of these phonemes, it was observed that all voiced segments' energy was above -75 dB, and towards boundaries, the energy drops slightly below the -75 dB threshold.

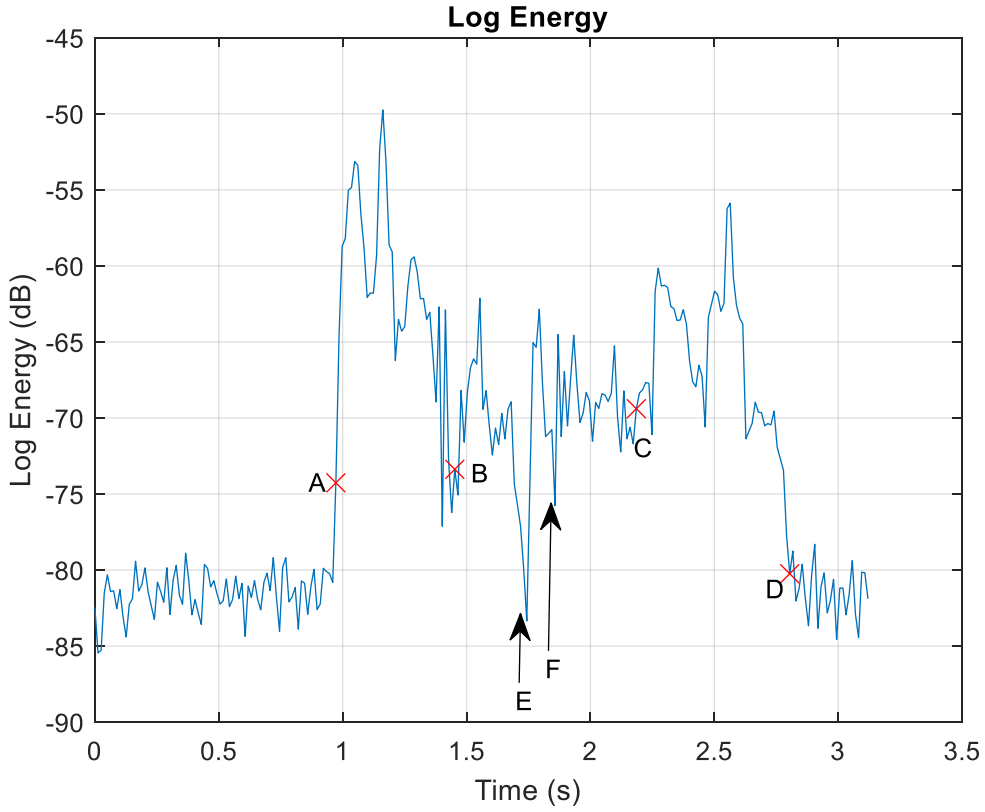


Figure 0-8: Baseline Utterance Log Energy

1.10.4 Zero Line crossing detector

The ZLC counts the number of zeros in each frame. Zero-crossings were detected if the conditions below were met:

$$x_i < 0 \text{ and } x_{i-1} > 0$$

$$x_i > 0 \text{ and } x_{i-1} < 0$$

where x is the sampled signal, x_i is the current sample and x_{i-1} is the previous sample. The number of zero-crossings was lower in the voiced segments and higher in and around the noisy frames. In word boundary sections, zero-line crossing increases significantly and sharply

decreases immediately after the sharp increase. The resultant zero-line crossing per frame for the baseline utterance is illustrated in Fig 2-9. Point A is the start of the utterance, point B and C are boundaries between words, the utterance ends at point D. Point E results from the unvoiced alveolar phone [th]. It was noted that in all voiced regions, the number of ZLC was lower than 50. Towards the second word boundary C, the number of ZLC rises slightly above the 50 ZLC threshold.

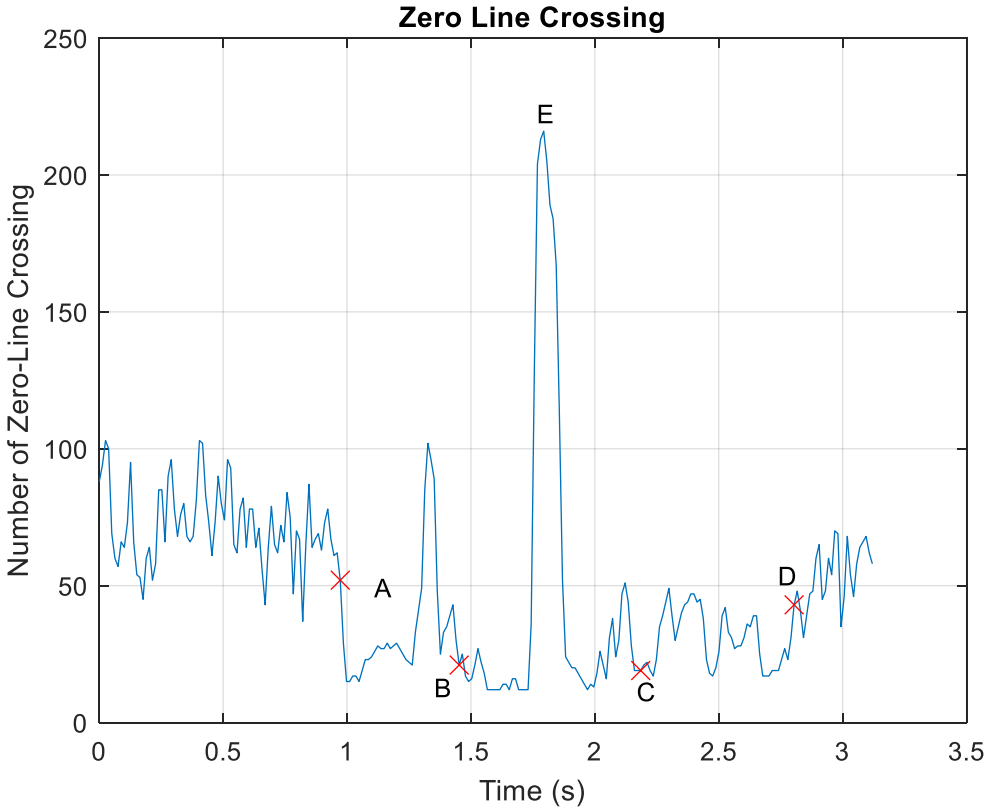


Figure 0-9: Zero Line Crossing Per Frame for the Baseline Utterance

1.10.5 Skewness

The skewness feature may be mathematically expressed as a HOS algorithm that computes the signal distribution asymmetry of the spectrum band energy. The spectral centroid is the mean of the signal spectrum as it represents the centre of gravity of the entire energy spectrum. A

skewness of zero indicates that the band energy is spread evenly on the centroid's right and left. Positive skewness indicates that the signal is spread more to the right of the centroid and negative skewness indicates a more leftward spread. Skewness is described by Eq. (2.9):

$$\widetilde{\mu_3} = \frac{E(x-\mu)^3}{E(\sigma)^3}, \tag{2.9}$$

where x is the time series, μ and σ are the mean and standard deviation of the observed signal frame, and $E(.)$ represents the expectation operator. The results in Fig 2-10 below suggest that frames with skewness between 0 and 3 are noisy, frames with skewness above 3 are voiced, and negative skewness indicates unvoiced regions. Notably, skewness at point E is negative due to the alveolar phoneme [th]. The coarticulation region – point F has 6 consecutive frames with skewness below the threshold of 3.

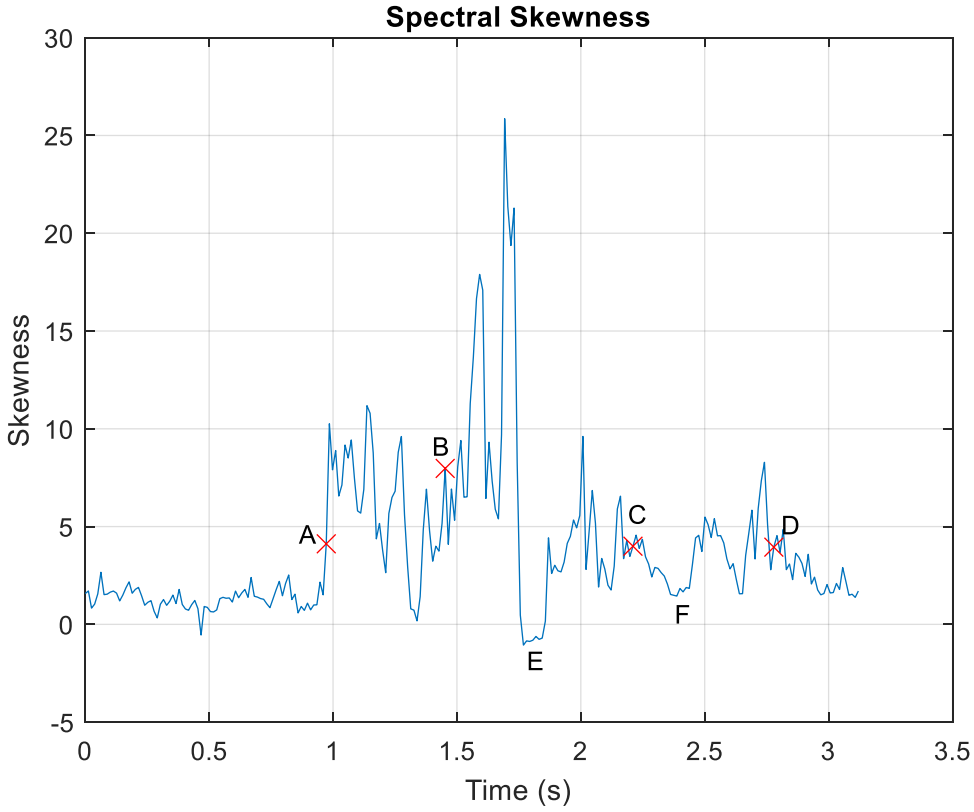


Figure 0-10:Spectral Skewness on the Baseline Utterance

1.10.6 Kurtosis

As with skewness, the kurtosis feature may also be mathematically expressed as a HOS algorithm, which computes the degree of flatness vs. peakedness of the frequency spectrum band energy. For a normal distribution, the kurtosis will be between 0 and 3; otherwise, the region is more prone to outliers, where negative and positive kurtosis indicate a flatter and a more peaked distribution of energy around the centroid. Kurtosis is described by Eq. (2.10):

$$Kurt = \frac{E(x-\mu)^4}{E(\sigma)^4} \tag{2.10}$$

where x is the time series, μ and σ are the mean and standard deviation of the observed signal frame, and $E(.)$ represents the expectation operator. The results of the spectral kurtosis of the baseline utterance are presented in Fig 2-11 below. These results suggest that the energy is distributed evenly in and around noisy regions. Voiced regions have a more peaked distribution. It can also be noted that the kurtosis drops towards zero around boundaries and increases sharply afterward. As it has been a trend in other features, points, E and F are an exception from the norm.

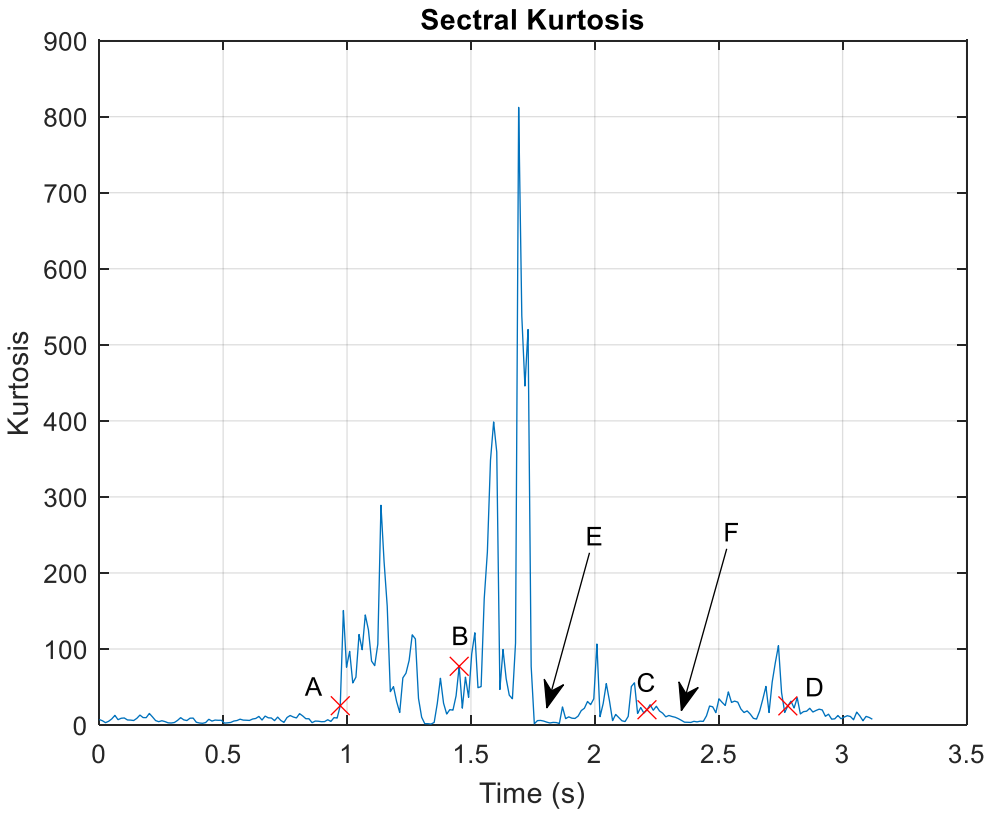


Figure 0-11:Spectral Kurtosis on the Baseline Utterance

1.10.7 Implementation

All features were implemented using the MATLAB audio toolbox. The code is attached in Appendix A. Features were evaluated for a sample of each utterance as listed in table 2.1. Results are illustrated from Fig. 2-12 to 2-18.

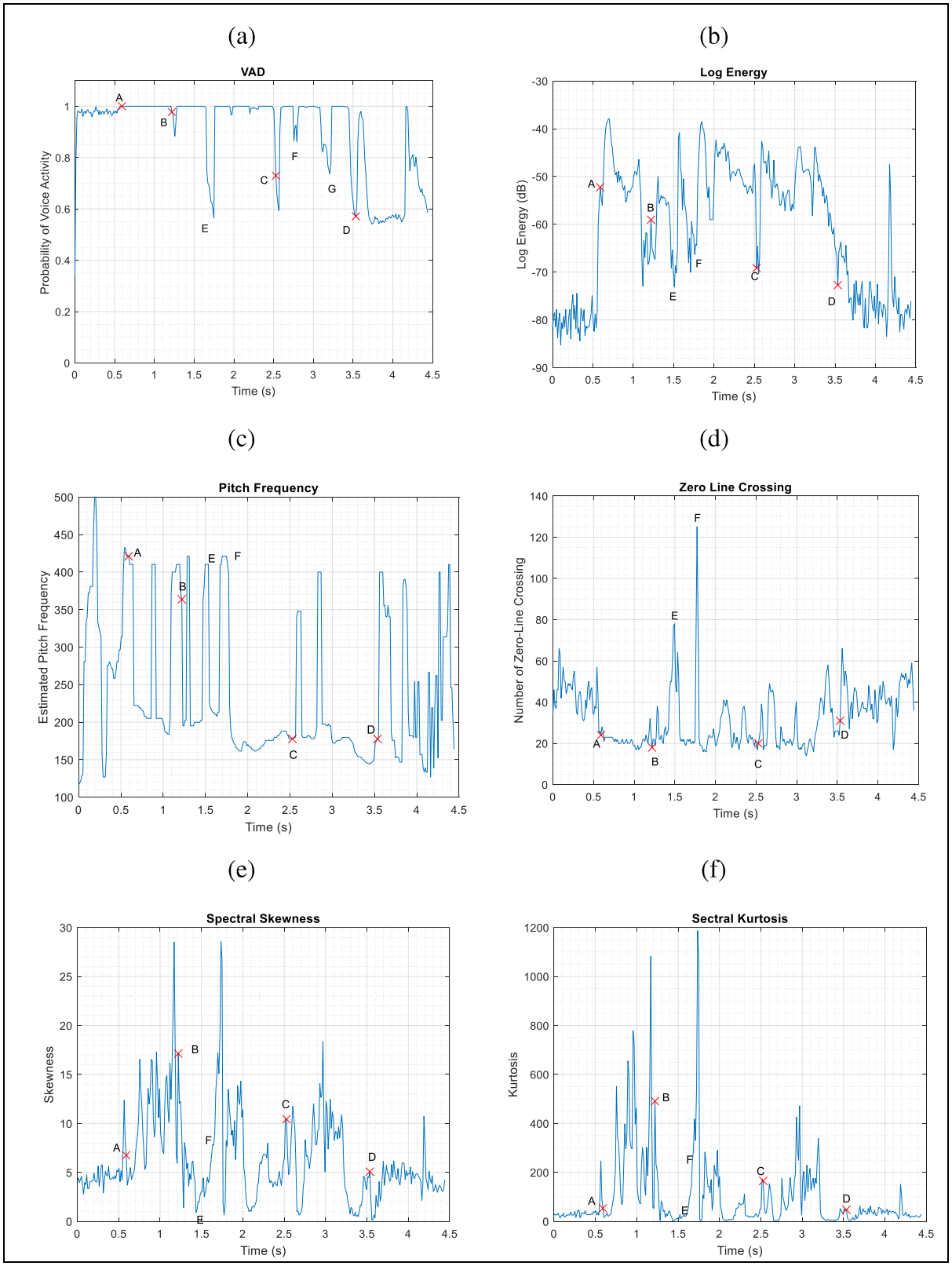


Figure 0-12: Boundary Feature Evaluation for "ehhovisi eliseduzane labezindaba"

Fig 2-12 above has features of an utterance from a 39-year-old female, hence the higher pitch frequency. Boundary regions were marked at the first syllable of each word. Points A, B, C, and D were as expected, following the boundary feature trend as previously established. Frames around points E and F are voiceless due to the semi-voiced approximant [lé] followed by the unvoiced fricatives [s]. While VAD, log energy and ZLC suggest that these points are two separate boundaries, skewness and kurtosis indicated them as one extended boundary. This may be used as an indication of an unvoiced segment within the word.

The utterance "iphepha lokuqala p" was recorded by a 19-year-old female. The results of its boundary features evaluation are illustrated in Fig 2-13 below. Boundary points A, B, C, and D were as expected as the boundary feature trend. The results particularly highlighted the effects of voiceless explosive velar [k′] at point E, the voiceless palatal click [ǃ] (q) at point F, and two voiceless explosive bilabials [ph] at points G and H. The effects of [k′] and 'ǃ' are more prominent in the VAD and pitch frequency in Fig 2-13 (a) and (c). The probability of voice activity is lower than 95% for both points, and the pitch frequency is higher. The effect of [ǃ] is further illustrated in figures 2.13 (d), (e), and (f), i.e., ZLC, spectral skewness, and kurtosis, respectively. Point F is 20 ZLC above the threshold of 50, skewness and kurtosis indicate that the spectrum is evenly distributed around the centroid, and the distribution is flat. The effects of the voiceless explosive bilabial, points G and H are illustrated in Fig 2-13 (d), (e), and (f).

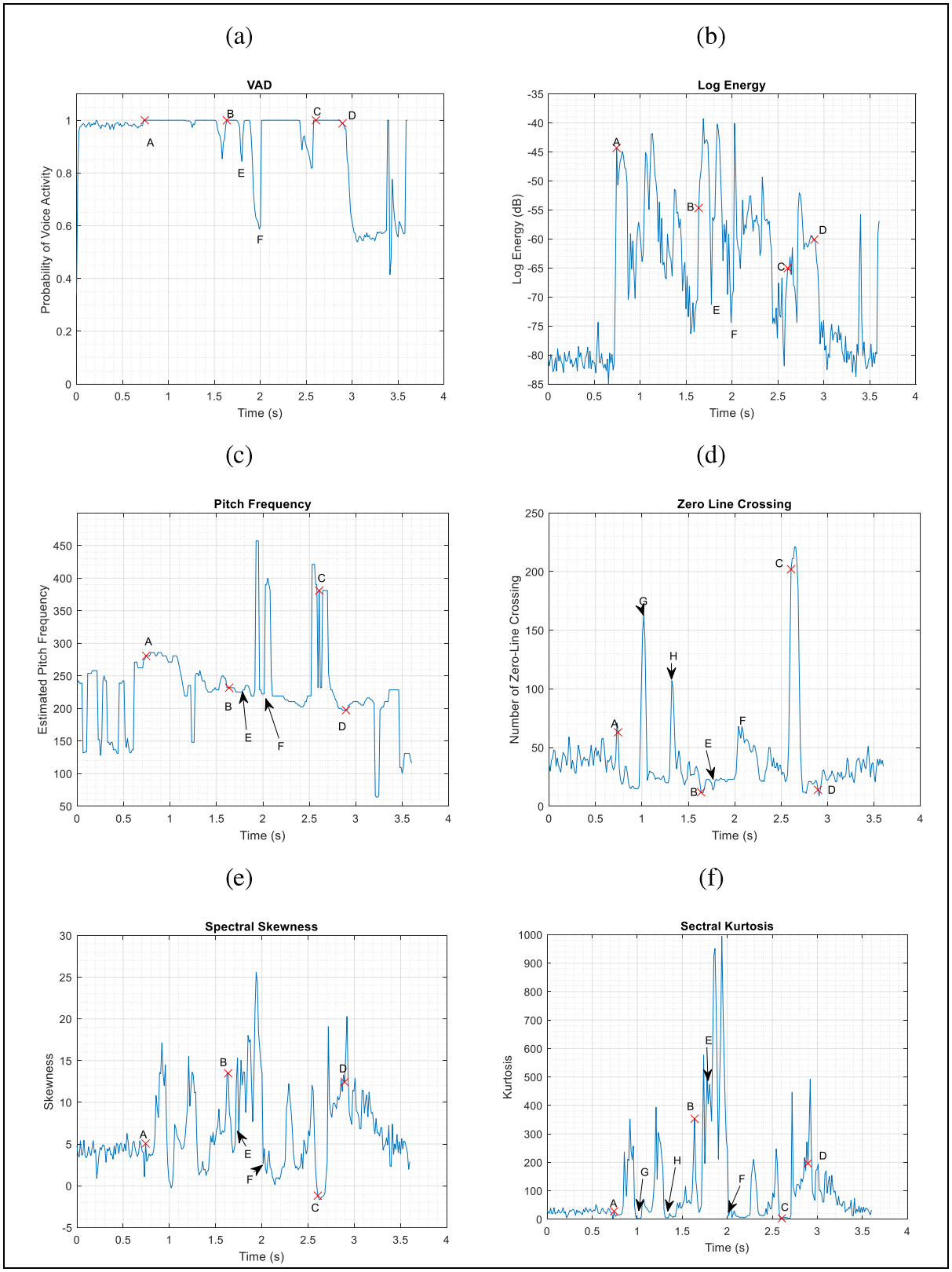


Figure 0-13: Boundary Feature Evaluation for "iphepha lokuqala p"

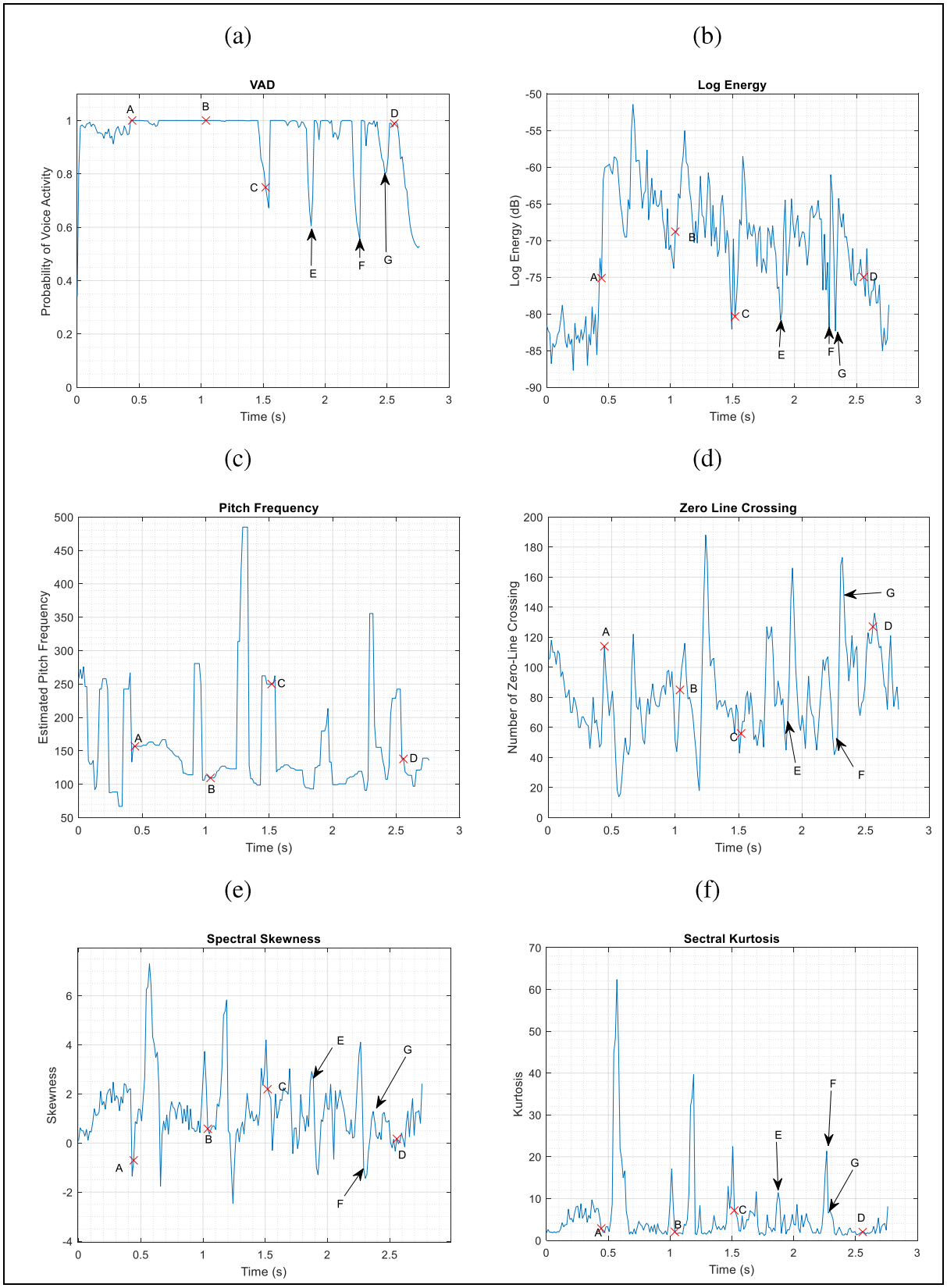


Figure 0-14:Boundary Feature Evaluation for "intela ngendlela evikelikele"

Boundary feature results for “intela ngendlela evikelikele” are illustrated in Fig 2-14. A 23-year-old male recorded the utterance. Boundary points A, C, and D were as expected. The features of boundary B were similar to those in the voiced regions; this is a result of the voiced lateral liquid alveolar [l] from the end of the first word extending to the voiced explosive velar. The outlier E is the result of the voiced fricative dentilabial [v]. Outliers F and G are both results of the unvoiced velar [k].

Fig 2-15 below illustrates the result of the utterance “abantu abafuna isondlo”, recoded by a 35-year-old female. Boundary features for points A, C, and D were as expected. At point B, the last vowel of the first word is [u], and the first vowel of the second word is [a]. These two vowels are pronounced from the velum; [u] is pronounced with the space between the tongue, and the soft palate is open and [a] is pronounced with the space closed. While transitioning from the first word to the second, the frequency and energy remain the same. At point E, there is the unvoiced dental fricative [s], the predeceasing vowel [i] is highly voiced, this affects the pronunciation. At point F, there is a voiced dental affricate [ndɧ].

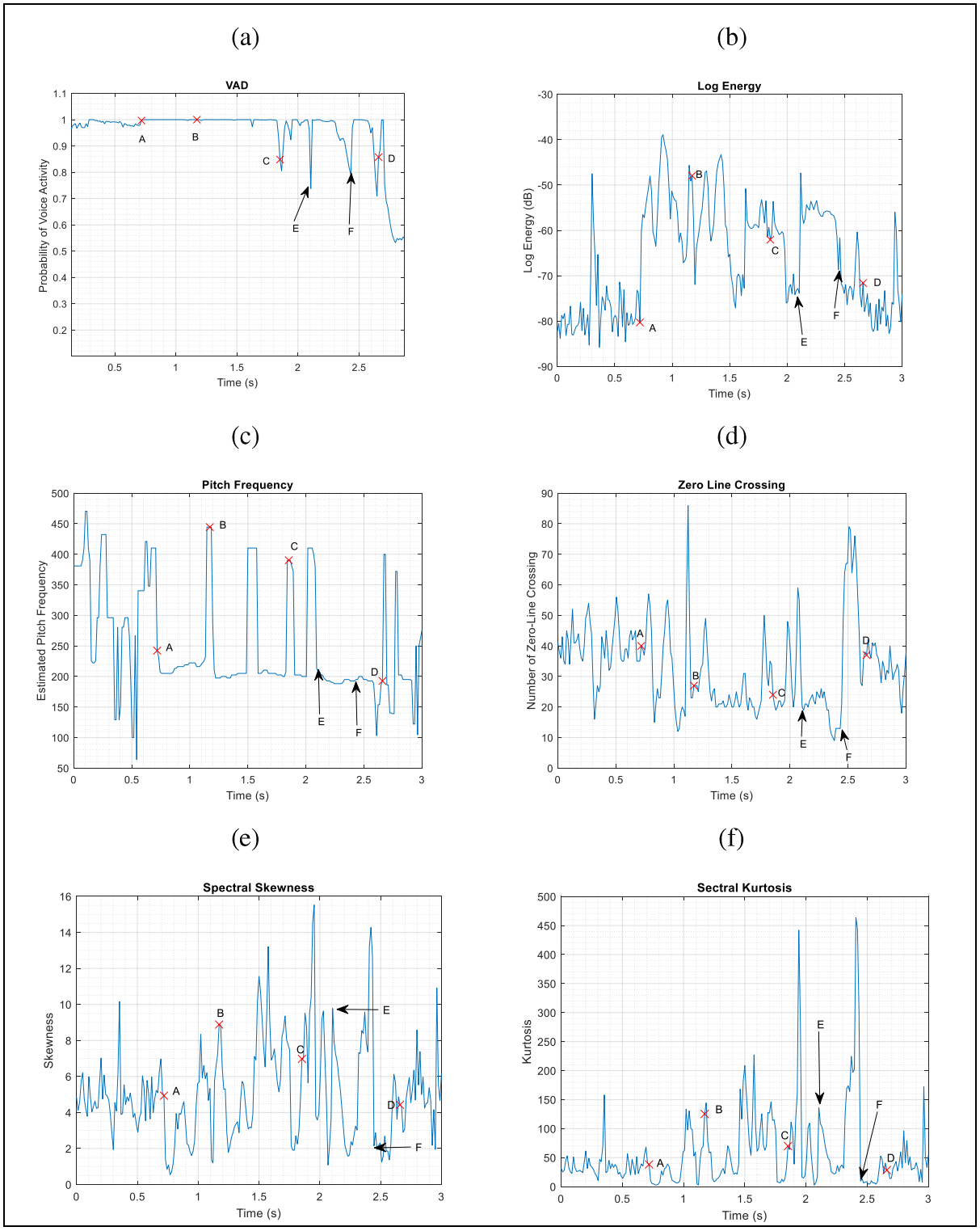


Figure 0-15: Boundary Feature Evaluation for "abantu abafuna isondlo"

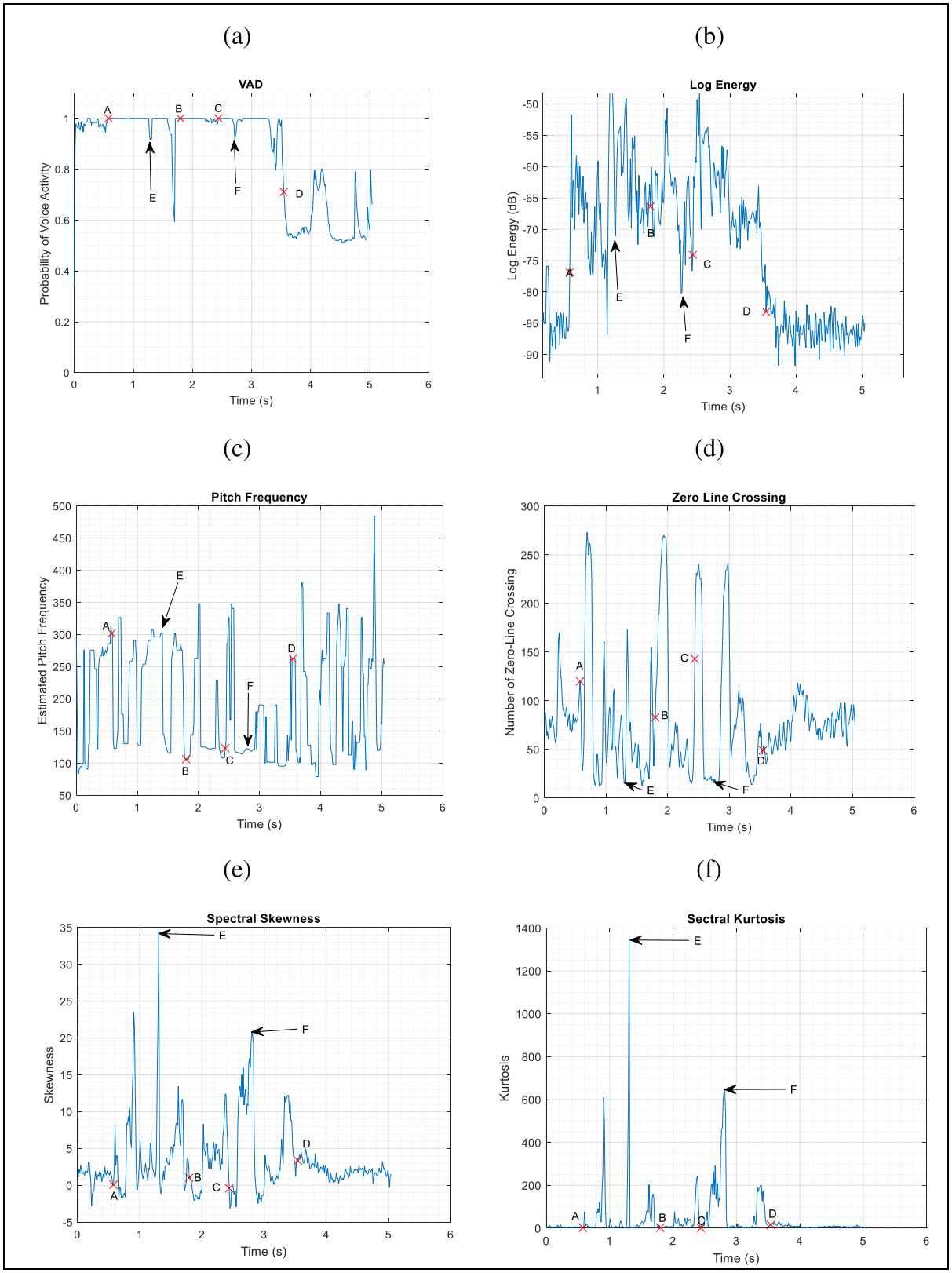


Figure 0-16:Boundary Feature Evaluation for "istifiketi sakho sokuzalwa"

Fig 2-16 above illustrates boundary feature evaluation for “isitifiketi sakho sokuzalwa” recorded by a 35-year-old male. All boundary points follow the same trend and are in line with the expected trend as previously discussed. Outliers point E and F were both a result of the unvoiced velar [k].

These experiments indicated the voiced segment threshold of certain features, the effect of unvoiced phonemes as well as causes of coarticulation. The minimum threshold for log energy was -75 dB, the maximum was -50 dB for males and -40 dB for females. Zero-line crossing ranged between 10 and 50 for females and 20 to 120 for males. While the voiced segment of male utterances had both negative and positive skewness, females had positive skewness. However, the trend regarding boundary skewness remained the same range of 0 to 3. Kurtosis had no particular threshold; the difference between the consecutive frames determined the boundaries.

1.11 Sentence Segment Classification

Boundary segments are a small portion of a sentence with voice segments having a bigger representation; this is shown in Fig. 2-17, where only 9% of the training data represent boundary segments. Most recognition algorithms favour the majority class where the training data is imbalanced. Two types of algorithms have been proposed as solutions to imbalanced training data; these are random oversampling and random undersampling (RUS). Both methods have their advantages and disadvantages.

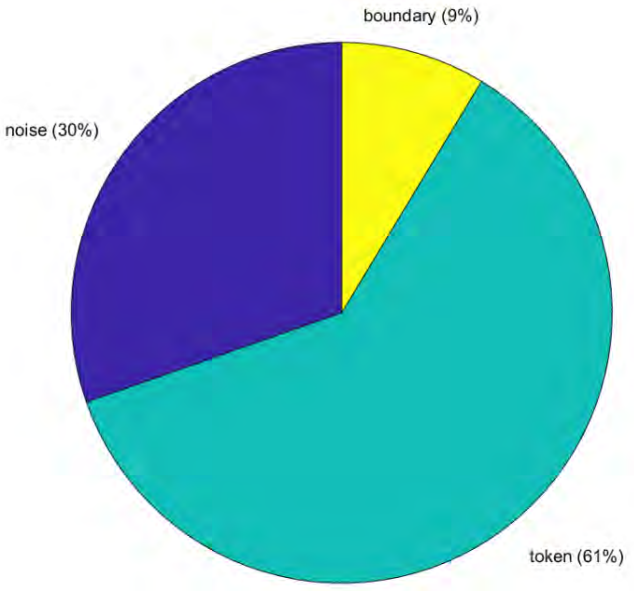


Figure 0-17: Training Data Frame Category Representation

Random oversampling duplicates the minority samples until the dataset is balanced. By oversampling, no data is lost. However, the increased number of samples increases the training time and may cause overfitting [35]. Similarly, RUS randomly deletes random majority samples, which means some data will be lost. However, by reducing the number of samples, the model training time is also reduced. Moreover, RUS has a higher sensitivity compared to oversampling [35]. For this reason, RUS was favourable for this research.

According to Naganoor et al [25], an ensemble classifier method had the best performance in their proposed approach. This research proposes a tree ensemble, which is a collaboration of tree-like decision making structures. Each decision tree is assigned a particular hypothesis. The final prediction is a result of majority voting. In this study, the adaptive boosting (AdaBoost) technique was used to create a tree ensemble. The combination of RUS and AdaBoost is known as RUSBoost [36].

The RUSBoost algorithm is split into three steps:

Weights of examples are initialised to $\frac{1}{m}$, where m represents the number of examples in the training dataset.

T weak hypotheses are iteratively trained as follows:

1. Most samples are randomly removed to achieve the desired class ratio. Resulting in a temporal training set S'_t with weight distribution D'_t
2. S'_t and D'_t are passed through the base leaner, which will generate a weak binary hypothesis h_t
3. A pseudo-loss ϵ_t is calculated using the equation below:

$$\epsilon_t = \sum D_i(t)(1 - h_t(x_i, y_i) + h_t(x_i, y)) \tag{2.11}$$

4. The weight update parameter $\propto$ is calculated as:

$$\propto = \frac{\epsilon_t}{1 - \epsilon_t} \tag{2.12}$$

The weight distribution for the next iteration is updated and normalised.

These steps are repeated until the T^{th} iteration.

5. The final hypotheses $H(x)$ is generated as a weighted vote of the T weak learners

$$H(x) = \underset{y \epsilon Y}{\text{argmax}} \sum_{t=1}^{T} h_t(x, y) log \frac{1}{\alpha_t} \tag{2.13}$$

Given (x_1, y_1) ... (x_m, y_m) where x is the feature of the signal and y is the class label; x_i is a point in feature space X, y_i is a class label in a set of labels Y . Each dataset (x_i, y_i) represent

the m^{th} example in dataset S, t is an iteration between one and the maximum number of iterations T.

This classification method was implemented using the MATLAB machine learning toolbox. Prosodic and HOS features, as discussed in section 2.5, were the classification variables.

1.12 Results and Discussion

The experiments were conducted taking into consideration the manually measured boundary segment averaged at 94 ms. Considering the window length of 20 ms (320 samples) and an overlap of 50% (160 samples) each frame was 10 ms long. Therefore, a predicted boundary was considered correct if it was within 10 frames of the actual boundary, to accommodate the 94 ms word coarticulation.

A comparison of the predicted class and true class was used to measure the performance of the proposed approach. Table 2.3 presents the segment recognition of the proposed approach and shows the accuracy trends observed in the experiments of each sentence. Accuracy was calculated from each sentence's independent confusion matrix using equation 2.14.

$$Accuracy = \frac{True\ Positives + True\ Negative}{the\ sum\ of\ observations} \tag{2.14}$$

Table 0.3. Word Boundary Classification Test Results Table

Sentence	Noise	Boundary	Token
“ehhovisi eleseduze labezindaba”	74.1%	51.5%	72.2%
“iphepha lokuqala p”	88.1%	66.6%	67.7%
“intela ngendlela evikelekile”	85.2%	58.5%	75.9%
“abantu abafuna isondlo”	78.2%	75.4%	74.0%
“isitifikedi sakho sokuzalwa”	85.9%	77.3%	74.9%
Average Accuracy	**83.4%**	**68.6%**	**70.1%**

It was noted that noise has the highest recognition accuracy in all sentences. This is mainly due to the distinct nature of the noise segment, whereas the distinction between the token and boundary is not as clear.

Boundary recognition for “ehhovisi eleseduze labezindaba” is the lowest due to the two semi-voiced approximants [lé] followed by the unvoiced fricatives [s]. Features had classified these segments as boundaries. During training, these segments were marked as tokens. Hence, 30.1% of boundaries were misclassified as tokens. Token recognition for “iphepha lokuqala p” is low due to the voiceless sounds [k´], [!], and [ph]. These phonemes have boundary segment features; hence 27.5% of token segments were classified as boundary segments. It was also

noted that boundary segments were equally misclassified as tokens. The utterance “intela ngendlela evikelekile” has a significantly low boundary classification compared to token segments. This may be caused by the voiced fricative dentilabial [v] followed by two unvoiced consecutive velars [k]. Moreover, the boundary between the first word and the second word is unclear due to the voiced lateral liquid alveolar [l] from the end of the first word extending to the voiced explosive velar [ŋg]. The token and boundary features of utterances “abantu abafuna isondlo” and “isitifikedi sakho sokuzalwa” were equally matched.

The overall results as illustrated in Fig 2-18 are encouraging as the recognition of token segments and boundary segments are closely matched at 70.1% and 68.6%, respectively. Moreover, the misclassification of the two segments to one another are closely matched. These results were anticipated due to the unclear distinction between the two segments

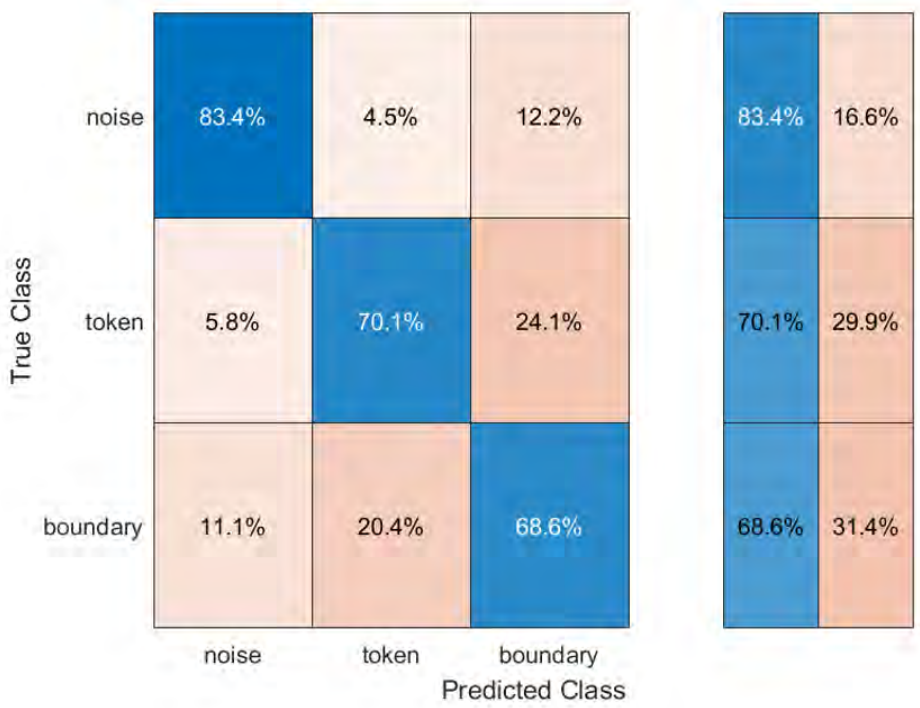


Figure 0-18: Confusion Matrix Test Results

The confusion matrix shows that 24.10% of tokens were incorrectly recognized as boundaries. This confusion is minimized by implementing a next boundary region threshold. Manual token duration measurements showed that a token in the dataset took more than 100 ms. Therefore, a boundary predicted within 100 ms or 10 frames after the previous boundary region was ignored.

1.13 Conclusion

Results achieved with the combination of prosodic and HOS features proposed in this study were lower than Hindi language boundary estimation [24]. This can be attributed to the training and testing data used; the latter had 3 speakers, each recording 51 sentences, while this study used data from an average of 33 speakers per sentence. The database structure in [24] had limited voice tones, and pitch frequencies, a third of the database had a similar pitch pattern. This experiment was not as speaker robust as in this study.

Naganoor *et. Al* [25] used the same approach proposed in this study on a bigger dataset with a large vocabulary. The quality and quantity of the data set made it a good candidate for use in the lexical analyzer, hence the superior results.

The results considered the automation of boundary recognition. The use of rudimentary and HOS features with recognition automated using the RUSBoots classifier produced encouraging results, with boundary region recognition of 68.6%. The dataset used was diversified such that the approach was not biased. For each of the 5 sentences used, individual confusion matrices were plotted. Boundary recognition ranged from 51.5% to 77.3%.

FEATURE EXTRACTION

1.14 Introduction

Feature extraction transforms the analogue speech signal to a mathematical representation, which could be understood by a digital system. The presentation should accentuate the relevant characteristics of the words or phonemes being processed. Two main concepts are used in the development of feature extraction. The first is the production method, which forms the basis for linear predictive coding (LPC). The second is the human perspective of the signal, which forms the basis for PLP.

Feature extraction is investigated in this chapter by examining both the production and human perspective concepts. The effectiveness of the features was evaluated through a simple classification method. Section 3.2 documents work related to the concepts. Section 3.3 presents the proposed feature extraction model for isiZulu ASR. Results are presented and discussed in section 3.4.

1.15 Related work

Feature extraction provides the ASR process with the most informative parameters of the utterance; this is one of the most important tasks in accurately recognising the speech signal [37]. Several feature extraction techniques have been developed including, retrieval based prosodic features, Wavelet Cepstral Coefficients (WCC), Linear Predictive Coding (LPC), and Mel-frequency Cepstral Coefficient (MFCC) [38].

Retrieval based prosodic features are largely used in determining the speaker role and speaking style. Prosodic features include average speaking and articulation rate, various F_0 statistics (mean, median, minimum, maximum, variance, and slope), and minimum, maximum, and mean (root mean square) RMS energy. Valente and Motlicek [39] proposed the use of prosodic features to extract speaker style. Prosodic feature extraction requires speaker based files with time aligned words and phone level transcript [40]. Prosodic feature extraction has also been used for sentence boundary classification and more recently prosodic for automatic language detection in [41].

MFCC and WCC and are cepstral analysis techniques. The former uses a DFT and the mel scale, whereas and the latter uses a discrete wavelet transform and a linear frequency scale. The mel scale is based on the human hearing perspective. The mapping between frequency in hertz and mel scale is linear below 1 kHz and algorithmic above 1 kHz; this is mathematically expressed by Eq. (3.1):

$$mel(f) = 2\,595\,\log_{10}\left(1 + \frac{f}{700}\right), \tag{3.1}$$

where f represents the frequency.

Adam *et al* [42] argue that WCC outperforms MFCC in both recognition accuracy and noise robustness. Their argument is based on WCC requiring fewer features than MFCC. However, MFCC is the most widely used feature extraction technique in speech, gender, and emotion recognition [43, 44]. MFCC attributes its popularity to its ability to simulate human hearing by using a nonlinear frequency unit [45, 46].

LPC [47] is based on the speech production concept, which is created by two sources i.e. lungs and vocal tract. The lungs and vocal cord produce a continuous signal with the vocal tract

acting as a filter. LPC attempts to recreate the two distinct components of speech $s(t)$; this is mathematically expressed by Eq. (3.2):

$$s(t) = n(t) * h_v(t), \tag{3.2}$$

where $s(t)$ is the convolution of the excitation signal represented by $n(t)$ with the vocal tract impulse represented by $h_v(t)$. LPC has produced a recognition accuracy of 81.7 % [38], compared to 93.5% by MFCC. In a separate experiment, a recognition accuracy of 99.04% by MFCC and 92.92% by using LPC is achieved [48].

1.16 MFCC feature extraction

In this research MFCC is employed, the system block diagram is illustrated in Fig. 3-1. The primary experiment was done using the full sentence "*ulwazi oluthile mayelana*".

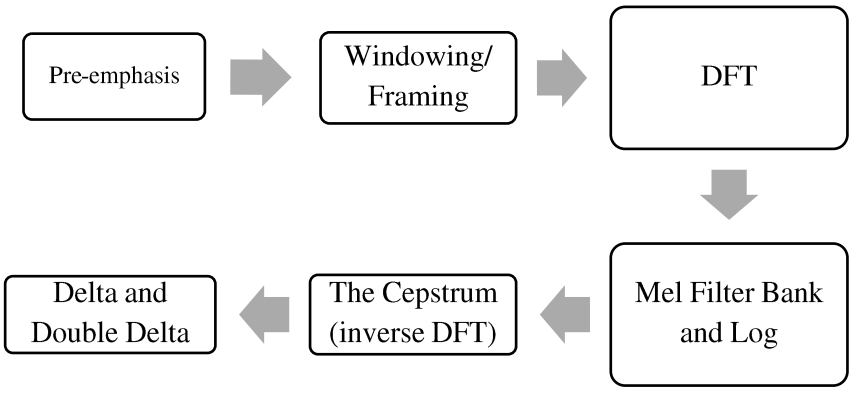


Figure 0-1 MFCC Block Diagram

1.16.1 Pre-Emphasis

This stage is implemented to boost higher frequency energy by applying a first-order high pass filter, which is represented by Eq. (3.1). The higher frequency boost avails acoustic model information from the speech to text formants found in frequencies above the maximum pitch 500 Hz and improves phoneme detection. As discussed in chapter 2, the frequency band 50 Hz to 500 Hz is the voice pitch range; the pitch is not required when extracting speech data relevant to ASR speech to text application.

$$y(t) = x(t) - ax(t-1) \tag{3.3}$$

$$\text{and } 0.9 \leq a \leq 1.0\ , \tag{3.4}$$

where y(t) is the resultant signal after the original signal $x(t)$ has been filtered. The results of implementing the above equation are illustrated in Fig. 3-2.

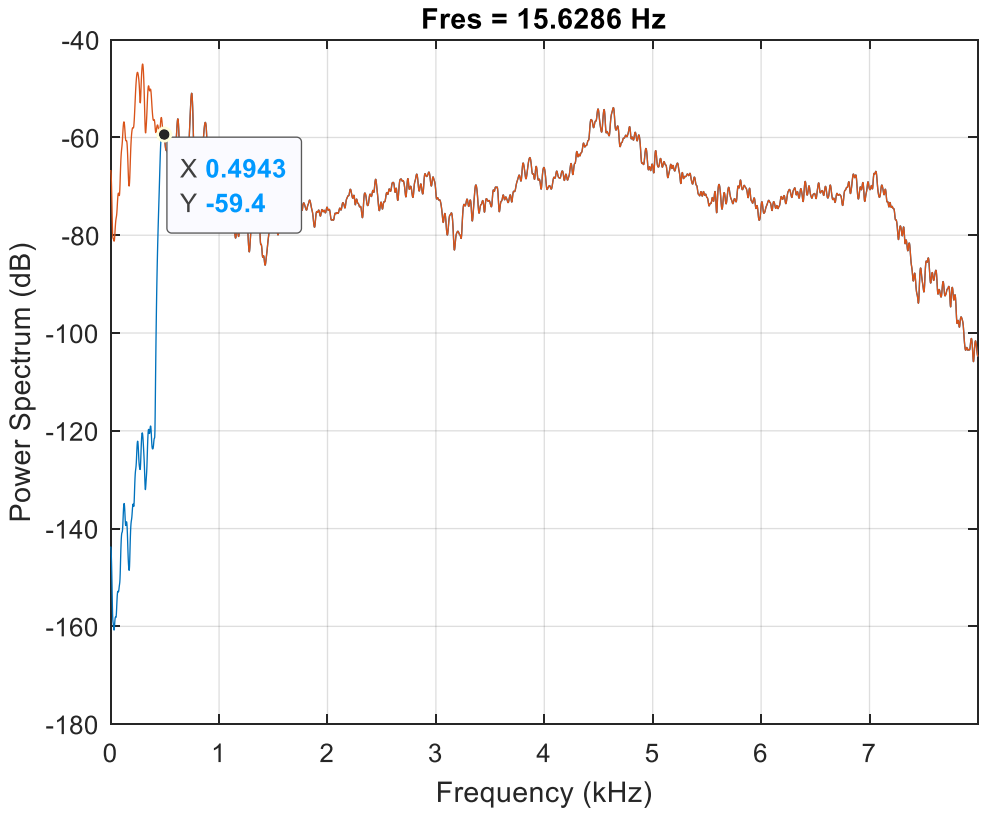


Figure 0-2 Pre-Emphasis Results

1.16.2 Window / Framing

Framing is implemented to segment the constantly changing audio signal. The two main windowing techniques are rectangular and hamming window. The former cuts off the frame in a straight edge; this is a challenge as the windows will be abruptly cut off. To avoid abrupt discontinuance between frames, as well as distortion in the spectrum, a Hamming window is applied. A Hamming window shrinks the values of the signal towards zero between window boundaries; this function is expressed by Eq. (3.4).

$$\omega(t) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi t}{L}), 0 \leq t \leq L-1 \\ 0, otherwise \end{cases} \quad (3.4)$$

It takes approximately 20 – 30 ms for a vocal tract to change its size and shape [49]. Hence, this is the standard framing range in ASR. In the proposed approach, the sampled signal was segmented in windows of 20 ms with an overlap of 50% for all feature extractions.

1.16.3 Discrete Fourier Transform

DFT is used to determine the frequency spectrum information for the windowed signal, allowing the system to detect how much energy is contained in each frequency band. The original signal spectrum and pre-processed signal spectrum are compared in Fig 3-3 and 3-4, respectively. A basic Fourier transform equation is used to obtain the framed signal's spectral information; this is mathematically expressed by Eq. (3.5):

$$X(k) = \sum_{t=0}^{L-1} x(t) e^{-j\frac{2\pi}{L}kt} \quad , \tag{3.5}$$

where L is the length of the signal and $X(k)$ is a complex number representing the magnitude and phase.

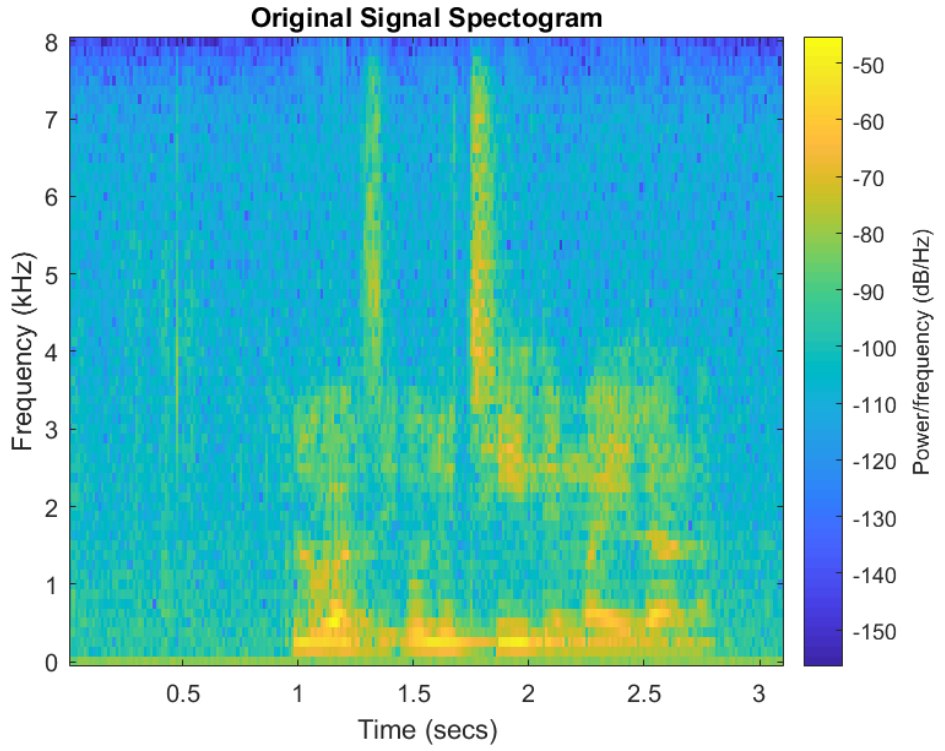


Figure 0-3: Original Audio Signal Spectogram

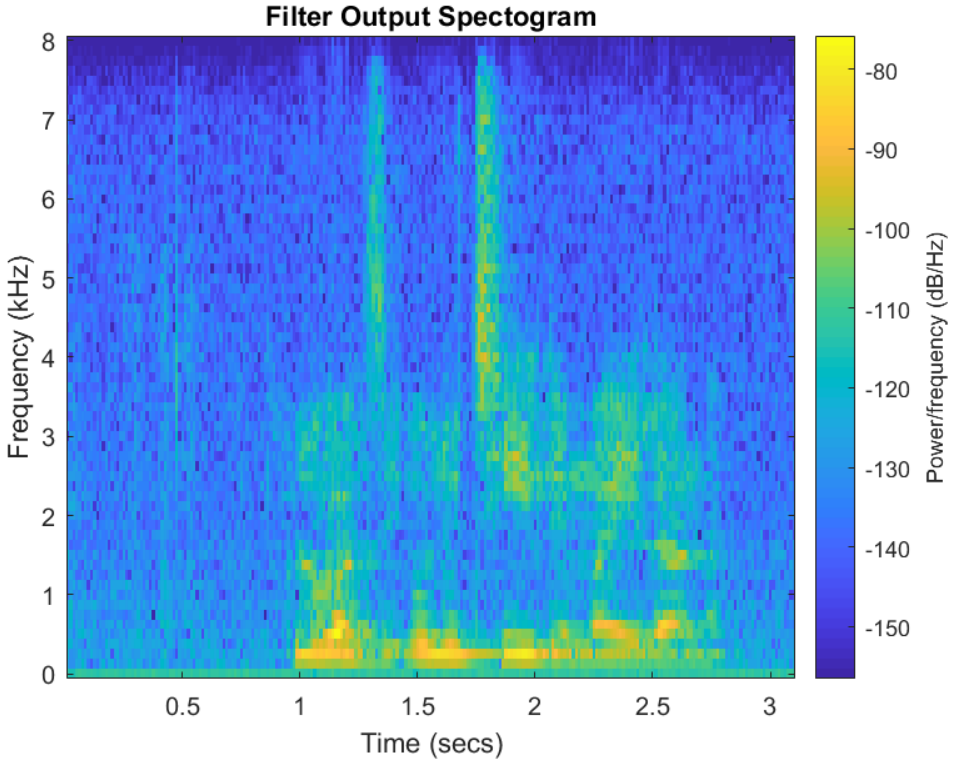


Figure 0-4: Filtered Signal Spectogram

1.16.4 Mel Frequency Filter Bank and Log

The output from the DFT provides information about the energy at each frequency band. However, human hearing is less accurate above 1 kHz. The MFCC model wraps the output of DFT onto the melody (Mel) scale. The Mel scale is an equidistant pitch separated by and equal number of Mel banks [22]. Mel filter banks were created to store the energy level at each frequency band, with 10 filter bins spaced linearly below 1 kHz and the remainder spread logarithmically; this is shown in Fig. 3-5. The mapping between hertz and mel scale frequency is linear below 1 kHz and algorithmic above 1 kHz as represented by Eq. (3.1) and Fig. 3-6.

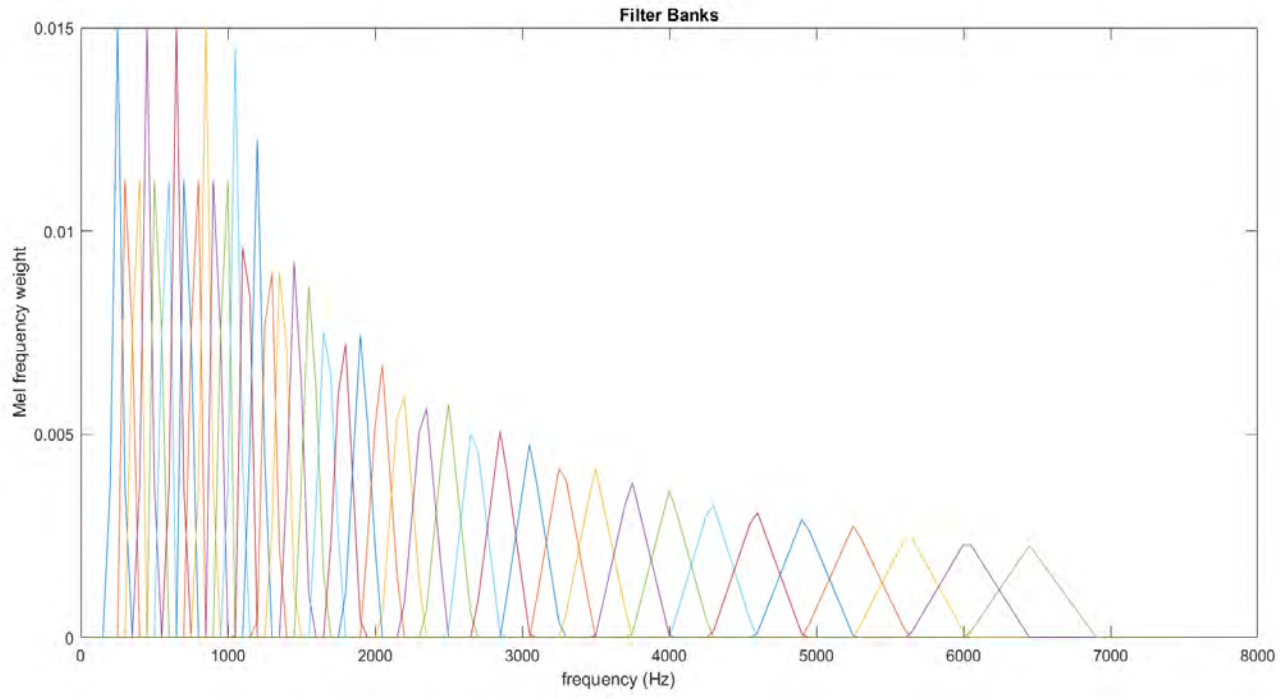


Figure 0-5: Mel Frequency Filter Bins

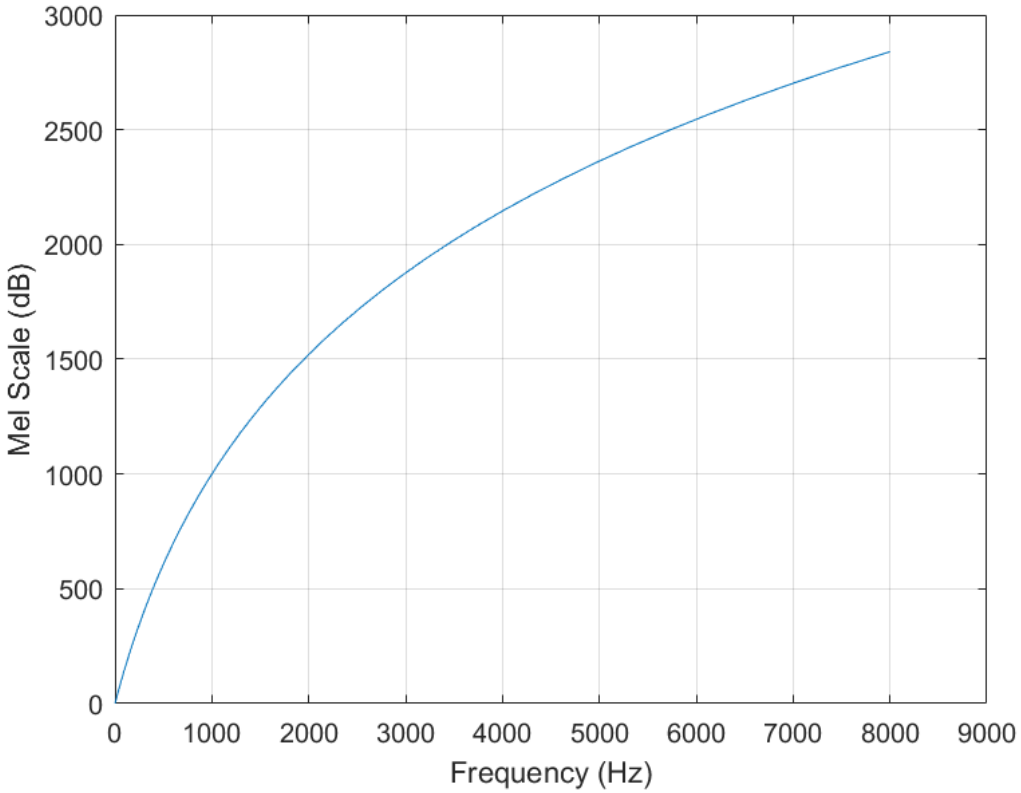


Figure 0-6: Mel Frequency Scale

1.16.5 The Cepstrum

Cepstrum is the spectrum of the log of the spectrum. It is more formally considered the inverse DFT. The cepstrum for a windowed frame of speech $x(t)$ is defined by Eq. (3.6):

$$c(t) = \sum_{t=0}^{L-1} \log_{10}(|\sum_{t=0}^{L-1} x(t)e^{-j\frac{2\pi}{L}kt}|)e^{j\frac{2\pi}{L}kt} \tag{3.6}$$

The inverse DFT takes the structure of a time-domain signal, also referred to as a pseudo-signal. Note: the label cepstrum is taken from the word spectrum; the first 4 letters are reversed.

The log spectrum contains high-frequency components enveloped by the fundamental frequency, glottal source waveform.

1.16.6 Deltas

A speech signal is not constant from frame to frame. The nature of the change between frames can provide useful information about phoneme identity. For this reason, features related to the change of cepstral features over time are computed by adding delta features to each filter.

The change in cepstrum features over time is used to further increase recognition accuracy by adding the change in velocity and acceleration to each of the cepstral coefficients. This may be computed as the difference between the prior and subsequent frame and expressed by Eq. (3.7).

$$d(t) = \frac{c(t+1) - c(t-1)}{2} \tag{3.7}$$

1.16.7 Recognition

For testing the suitable number of features for optimum recognition, the K-nearest neighbour algorithm (k-NN) was used. It was chosen for its simplicity and low computational cost. k- NN is an instant-based and lazy learner algorithm. The algorithm's task is to search the training dataset for the k most similar samples [50]. For example, in a three-class, two feature case with $k = 7$, the algorithm will examine the closest seven training samples. The new instant (x_i) will be classified as the most frequently represented class in the 7 training samples [51]. This is

illustrated in Fig. 3-7, where class A, B, and C are possible classes for the new observation x_i. In this case $x_i$class B is the most represented class, hence, x_i will be classified as class B.

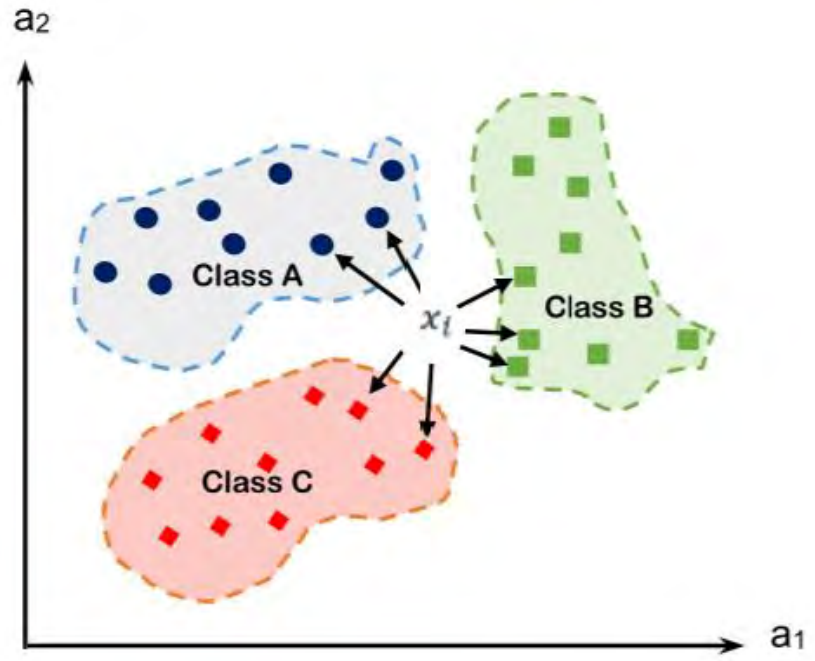


Figure 0-7: k-NN Classification Example Adapted From [51]

As a nonparametric algorithm, *k*-NN depends heavily on the distance matrix. Various distance measure matrices can be used in distance-dependent classifiers, including, among others, Euclidean, Manhattan, Cosine 50, and Mahalanobis [52]. Five isiZulu words were chosen to optimise the *k*-NN classifier. After hyperparameter optimisation, the Manhattan distance (also known as "city block" or "taxicab") and k=3 were chosen. The *k*-NN classifier was implemented using the algorithm from the MATLAB statistics and machine learning toolbox. The software determines the optimum number of neighbours among integers in the range $[1, \max(2, \text{round}(n/2))]$, where n is the number of observations. Each distance matrix was

tested to determine the optimum distance. The classifier was optimised using 30% of the test data.

The Manhattan distance is usually used in problems requiring distance between two points in a grid setting. The distance (d) is calculated as the sum of the absolute difference between the cartesian coordinates; this is mathematically represented by Eq. (3.8), where x_i and y_i are variables for the two points and i is the number of variables. In this section, the n is the number of MFCC coefficients.

$$d = \sum_{i=1}^{n} |x_i - y_i| \tag{3.8}$$

1.17 Results and Discussion

Words for this experiment were chosen according to their frequency in the NCHLT vocabulary. The five words listed in Table 3.1 were isolated from their original recording and stored in their respective audio data store. The isolation was done using the MATLAB audio labeller in conjunction with a MATLAB audio recording function. These words were chosen to compare the effect of the number of coefficients in recognition accuracy. Each word was cut out from 200 different sentences uttered by different people genders were equally represented, as shown in table 3.1. In each data store, 60% of the audio files were training data, and 40% were for testing.

In each simulation, the number of coefficients was changed starting from 9 to 17. The 5-fold validation results are presented in Table 3.2. There is a significant increase in accuracy from the simulation with 9, 11, and 13 coefficients. The accuracy difference between 13 and 15 coefficients is 1%. The difference between 15 and 17 coefficients is 0.67%. This experiment

concluded that coefficients above 13 did not contribute any significant improvement to word recognition accuracy.

Table 0.1: MFCC Test Data

Word	Females	Males	Total	The average number of frames (480 samples per frame and 60%)
Imali	99	101	200	29
Lomkhandlu	98	102	200	130
lukuqala	93	110	203	100
Ngendlela	100	100	200	100
Ulwazi	99	101	200	80

Table 0.2: MFCC Test Results

Number of Coefficients	5-fold accuracy Validation
9	60.08%
11	62.05%
13	63.14%
15	64.18%
17	64.85%

The confusion matrix in Fig 3-8 is the recognition results of 13 coefficient MFCC features. The confusion matrix represents the performance of the classifier model. For each of the words, the matrix shows four prediction parameters as listed below.

- True positive (TP): the number of predictions that the classifier correctly predicted the positive class as positive. Positive classes are the blue cell in the diagonal line of cells. For each class, there can only be one TP cell. In contrast, false-positive (FP) are negative classes predicted as positive. In this confusion matrix, these predictions are horizontally aligned with the TP cell.
- False negative (FN): the number of predictions where positive classes were classified as negatives. In this confusion matrix, these cells are vertically aligned with the TP. The opposite is true negatives, where negative classes are predicted as negatives. These are the cells that are not vertically or horizontally in line with the TP cell.

This experiment was conducted to determine the classification accuracy rate for each word, based on the recall and precision rates as defined in equations 3.9 and 3.10, respectively. These rates were used to the effectiveness of the feature as mentioned earlier extraction method for each word and make comparison among the words.

$$recall\ rate = \frac{TP}{TP + FN} \tag{3.9}$$

$$precision\ rate = \frac{TP}{TP + FP} \tag{3.10}$$

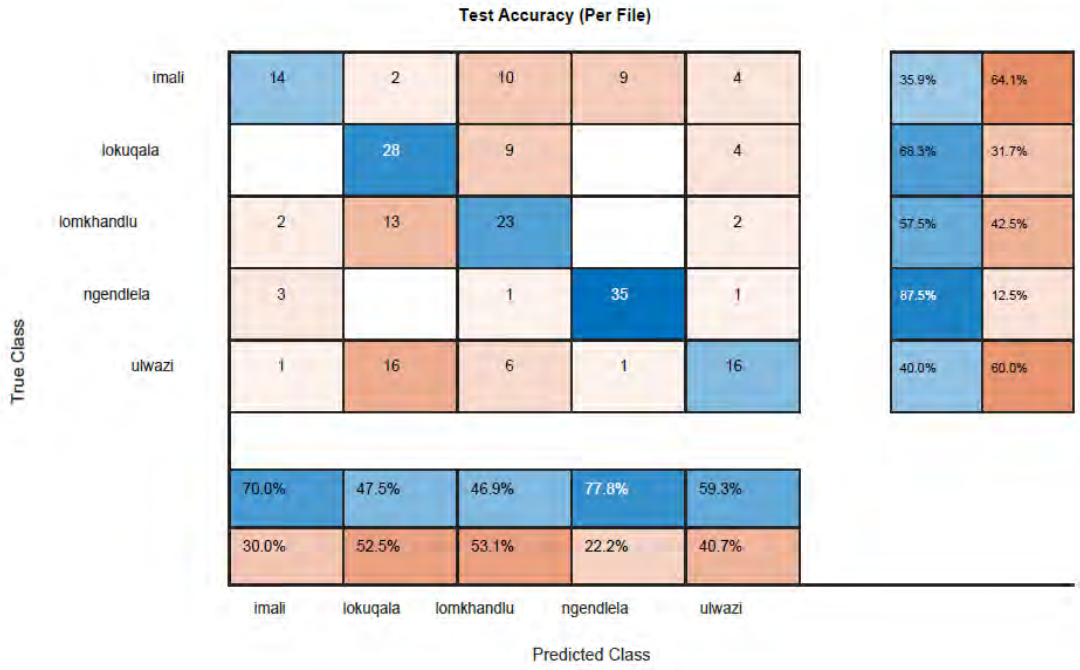


Figure 0-8:Test Validation per File

Table 0.3: Recall and Precision Rates Results Base on Word Feature Extraction

	Recall rate	**Precision rate**
"imali"	70.0%	35.9%
"lokuqala"	47.5%	68.3%
"lomkhandlu"	46.9%	57.5%
"ngendlela"	77.8%	87.8%
"ulwazi"	59.3%	40.0%

While the utterance "imali" has the lowest precision rate, it also has the second-highest recall rate. The difference between recall rate and precision is the number of FN and FP, respectively. Therefore, the number of FP is 58.6% higher than FN. The "Lomkhandlu" predictions had the highest number of incorrect classifications as "imali", hence the lower recall rate for

"umkhandlu". This may be caused by the repetition of speakers in the training data "imali" class; 11 speakers were repeated more than five times. For all those repeated speakers, the "umkhandlu" class had an average of 2 repetitions per speaker. The same is true for "ngendlela" incorrect predictions to "imali" class. However, the "ngendlela" class had more speaker repetition matched to those of "imali" class. It is also notable that the "ngendlela" class has higher recall and precision rates, attributed to low FN and FP, respectively. The highest contributor to FN is incorrect predictions to "imali" class. The training data for "lokuqala" and "ulwazi" classes matched the speaker repetition number of "imali" class; hence there are fewer FP from those predictions.

The "lomkhandlu" class has a significantly low precision rate due to the misclassification of "lokuqala" predictions. The mismatch of training data significantly affected recall and precision rates in all classes.

1.18 Conclusion

There are several word feature extraction for isolated word recognition application; this chapter discussed two dominant methods viz. LPC and MFCC. The latter was identified as the most favourable method based on feature extraction work that has been done in the field. After experimenting with 9 to 17 coefficients using *k*-NN as a classifier and 5- word database, 13 coefficients achieved the best results at 63.14%. Each coefficient contained cepstral, change in velocity, and acceleration, bringing the number of coefficients to 39. The mismatch in the training data affected the recall and precision rate. In future research it is recommended that a more evenly distributed speaker recordings be sourced.

1.19 Introduction

ML employs large amounts of data to train a computer (machine) to learn from past experiences to make future decisions. ML focuses on learning and adapting to new inputs with minimal human intervention. The process of machine learning has 5 stages as shown in Fig. 4-1.

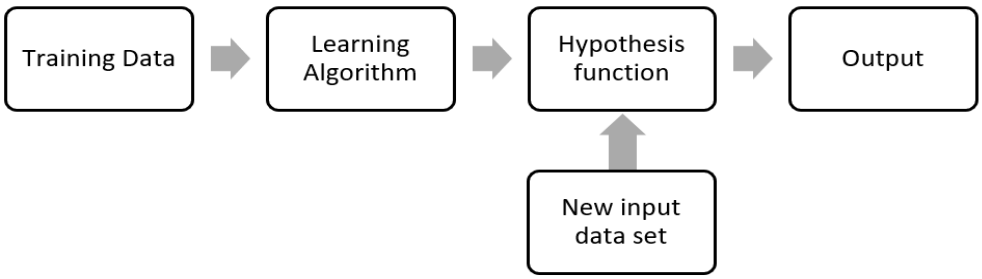


Figure 0-1: Machine Learning Stages

Machine learning has been employed in numerous applications over the years. Such applications include fraud detection, web search spam filter, credit scoring, and ad placements [53]. There are 5 learning techniques; these include supervised learning, unsupervised learning, semi-supervised learning, reinforced learning [54], and deep learning [44]. Each of these techniques is unique in its implementation; the hierarchy is shown in Fig. 4-2. The choice of which learning technique to use is determined by the nature of the problem and data size.

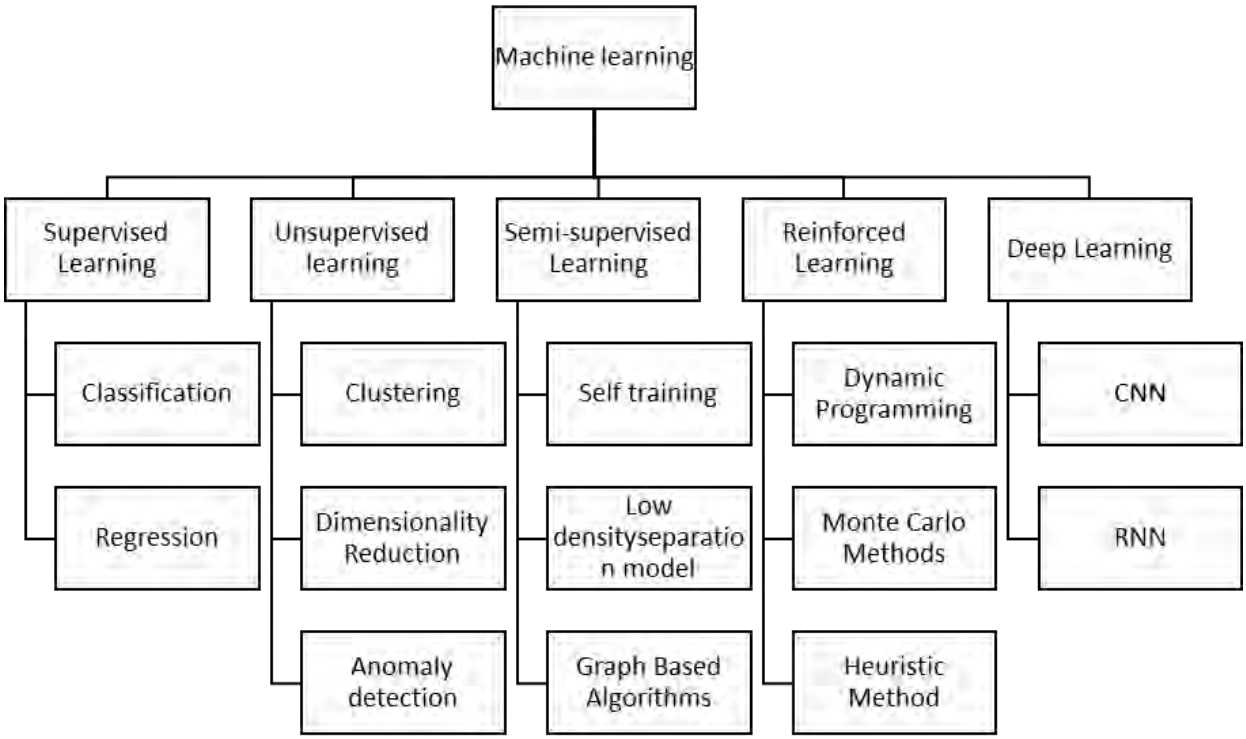


Figure 0-2:Machine Learning Types and Algorithms

1.19.1 Supervised learning

Supervised learning guides the learning process with the use of examples. An input with its target output is used to train the algorithm, these examples are then used to predict outputs of future inputs. The training input can be represented by a vector and its corresponding target output is described as a supervisory signal [55]. The learning algorithm is termed supervised learning as the target output is known and the algorithm attempts to reiterate the target output. This algorithm is best when there is limited training data.

1.19.2 Unsupervised learning

In this algorithm, unlabelled inputs are used for training. The algorithm extract statistically relevant information from the input [56]. The system analyses the input patterns to determine important information without external intervention. The main objective of unsupervised learning is to learn more about the structure and patterns in the data; the inputs are then clustered into groups based on the features extracted from the input objects.

Although this technique cannot name the resultant clusters, it may be used to produce differentiation between previously clustered and new inputs, which will allow for group clustering. With this method, foreign inputs will not be discarded but rather used to create a new cluster group where future inputs with similar patterns will be included. This learning will only work at its optimum when there is sufficient data.

1.19.3 Reinforcement Learning

Reinforcement learning is a consequential learning technique. The machine or agent is rewarded or punished based on its response to an instruction [57]. Reinforcement learning differs from supervised learning as it does not need an input/output pair but focuses on exploring uncharted areas and exploiting current knowledge. The learning process comprise controller, process, state, action, and the reward function as illustrated in Fig. 4-3. The process sends a state to the controller, the controller responds with an action, the reward function then compares the action with the predefined responses. It subsequently determines the reward or punishment for the controller.

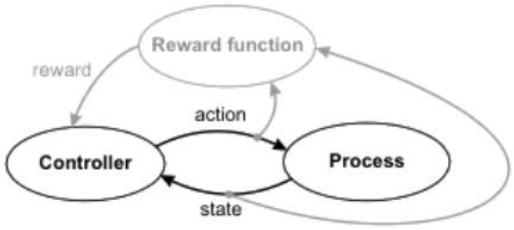


Figure 0-3: The flow of interaction in Reinforcement Learning

1.19.4 Semi-supervised learning

Semi-supervised learning combines supervised and unsupervised learning. A small amount of input/output paired training data and a larger amount of unlabelled data. Unlabelled data is clustered in relation to existing data distribution. When unlabelled data is clustered it is assumed that the data shares the same label as the existing labelled data. The main aim of this learning technique is to exploit a smaller amount of labelled data in order to label a larger amount of unlabelled data [29].

1.19.5 Deep Learning

Deep learning (DL) is inspired by human brain structure, functioning, and the capability to self-learn [58]. The neurological structure of the brain is imitated by ANN algorithms. DL research has been building on, and contributing to, many different research topics, such as feature learning, unsupervised learning, pattern recognition, and signal processing [59]. It has recently become a major subject of interest in the ASR community for its ability to improve acoustic modelling [60]. Training an ANN requires a large amount of data. Hence, the acoustic modelling improvement is only limited to a handful of languages with sufficient resources.

Research has shown that acoustic modelling for under-resourced languages can be improve by sharing hidden layers of an ANN algorithm across languages of similar structure. Making the hidden layers language independent, while the classifier layers are language dependent.

1.19.6 Chapter Summary

This chapter explores a few ML techniques that may be considered for use in an isiZulu ASR. While there are many ML techniques, this study compares 3 under the same conditions. Section 4.2 presents ML work that has been done in relation to ASR. Section 4.3 details the simulation experiments. Results comparison is presented in section 4.4, and finally, the chapter is concluded in section 4.5.

1.20 Related work

The influence of ML on ASR has increased over the years and ASR has conversely been a big topic of discussion and research in the ML society. A wide range of ASR algorithms have been used, but three have dominated the research community; these include HMM, SVM, and most recently, DNN. HMM was introduced for use in ASR in the mid-1970s and is still the widely in state-of-the-art ASR. The effectiveness of HMM is attributed to its left-to-right structure, the number of flexible states, and its ability to treat every sound separately [45, 61]. SVMs have yielded better accuracy compared to HMMs in real-time applications [62]. DNNs has recently become a major topic of research in the ASR community especially for under-resourced languages [60].

1.20.1 Support Vector Machine

An SVM is a dichotomic classification algorithm and for many years used for binary classification [63]. Although SVMs are considered dichotomic algorithms they may be configured to classify more than two classes [64]. A basic approach to multiclass SVM is a combination of N independent binary classification tasks [65]. The output code is defined by matrix R sized $N \times M$, where M is the number of classes and N is the number of tasks $r_{ij} \epsilon \{-1, 0, 1\}$. The commonly used output codes are: One-vs-All (OvA), One-vs-One (OvO) and Error-correcting code (ECOC).

i. *One-vs-All (OvA):* For M classes, there must be M classification tasks. For classifier $f^i(x)$, positive examples are points in class i and negative examples are not in class i ; this is illustrated in Fig. 4-4.

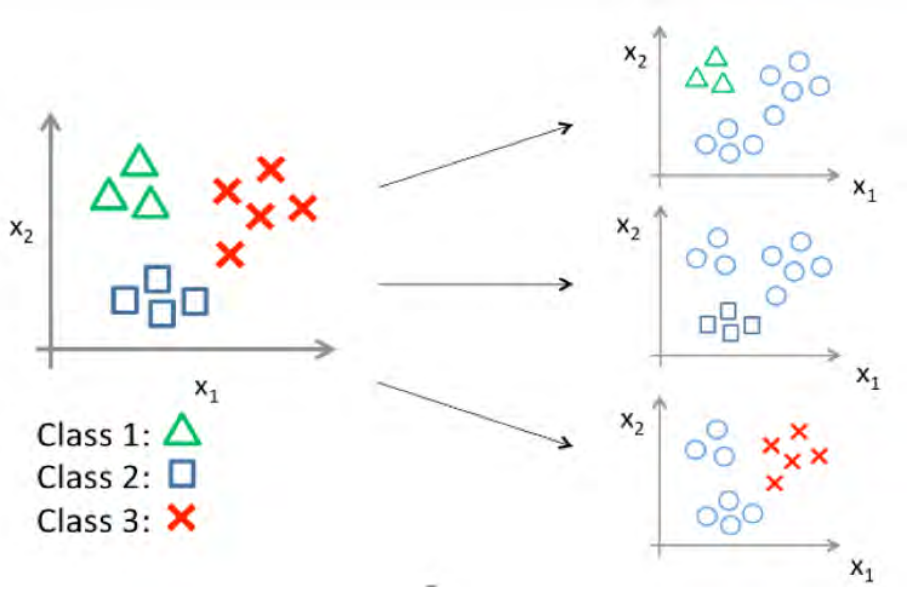


Figure 0-4: One-vs-All SVM Output Code Illustration adapted from [66]

ii. *One vs One (OvO)*: For M classes there must be $\frac{M\times(M-1)}{2}$ classifiers, where each classifier task has one class opposing every other class. Classifier $f^{ij}(x)$ distinguishes between class i and class j, where i has positive class members and j has negative class members; this is illustrated in Fig. 4-5. In this illustration classifier 1 compares classes B and C, classifier 2 compares classes C and B, and classifier 3 compares classes A and B.

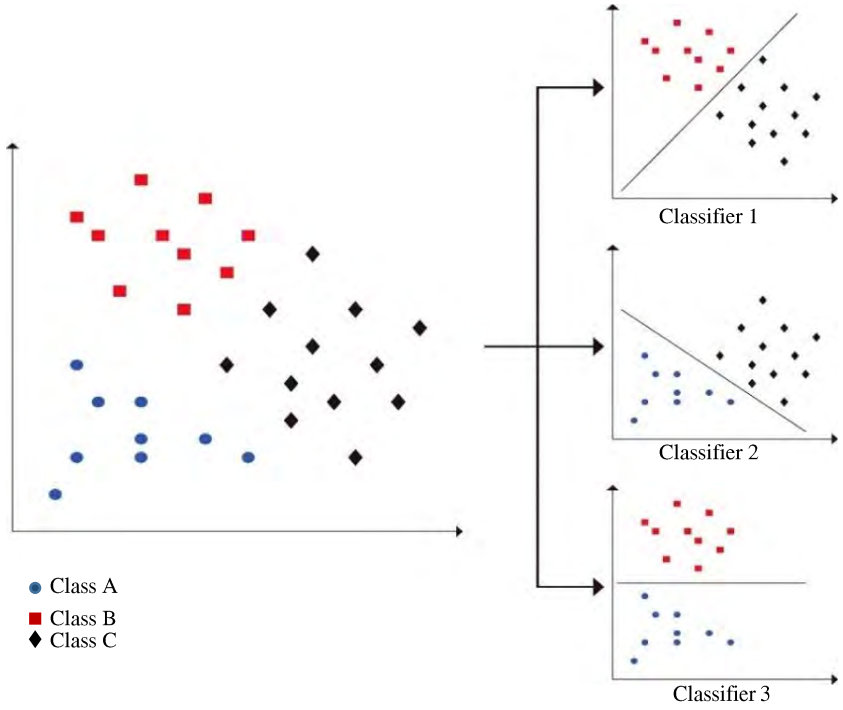


Figure 0-5: One-vs-One SVM Output Code Illustration

iii. *Error-correcting code (ECOC)*: This is used alongside OvO, enabling the SVM ensemble to reconstruct labels from noisy predicted binary labels [65]. The ECOC algorithm follows the following two steps:

1. Learner 1 is trained using class 1 or class 2 observations, where class 1 will be a positive class and class 2 a negative class. This is done for all learners.
2. For output code R with elements r_{ij} and s_j being the predicted classification score for the positive class of learner j. A new observation is assigned to class k that minimises the losses for the N learners. This is mathematically expressed by equation 4.2.

$$k = \underset{k}{\operatorname{argmin}} \frac{\sum_{j=1}^{N} |r_{ij}| \mathrm{g}\,(r_{ij}, s_j)}{\sum_{j=1}^{N} |r_{ij}|} \tag{4.2}$$

While SVM is extremely efficient in real-time applications, when margins are clear, the same cannot be said for high volumes of data, as the separation margins cannot be clearly marked. It is also inefficient when the data has noise [65]. A summary of ASR research applying SVM is tabulated in Table 4.1.

Table 0.1: Summary Of Automatic Speech Recognition Research That Specifically Uses Support Vector Machine For Classification

Classification technique	**Type of Speech**	**Reference**	**Number of Subjects**	**Hours of voice**	**Vocabulary Size**	**Word Classification Accuracy**
Hybrid SVM/HMM Systems	Continuous	[67]	3 146	50	24	86.0%
SVM and Minimum Distance Classifier	Isolated speech	[62]	4	not specified	40	90.6%

Ranking SVM	Isolated speech	[68]	Not specified	700	20	80.7%
Ranking SVM	continuous	[69]	Not specified	194	23 000	77.2%
SVM (hardware application)	Isolated	[64]	20	12 000 samples	30 words	98.5%

1.20.2 Hidden Markov Model

HMM is a statistical modelling tool popularly used in a wide range of time-series data. HMMs learn from previous activities between states and predict the most probable succeeding state. The activity of moving from one state to another is called a transition event. HMM most probable state M is defined by the set of N states, K observations, and three probabilistic matrices [70], this is mathematically represented by equation 4.3:

$$M = \{\Pi, \mathrm{A}, \mathrm{B}\}, \tag{4.3}$$

where:

$\Pi = \pi_i$ initial state probabilities

$A = a_{i.j}$ state transition probability

$B = b_{i.j.k}$ symbol emission probabilities

A state transition chain has three types of model topologies viz. ergodic, Bakis general left-to-right, and Bakis linear [70]. In the ergodic model, any state can succeed any other state; this is shown in Fig. 4-6. The Bakis general model is a “stay, move forward of skip one” model and the linear model is a “stay or move” model; these models are illustrated in Figs. 4-7 and 4-8, respectively.

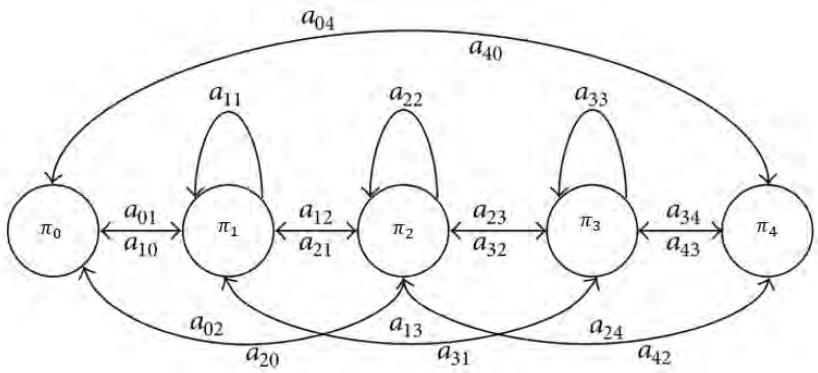


Figure 0-6: HMM Ergodic Structure [70]

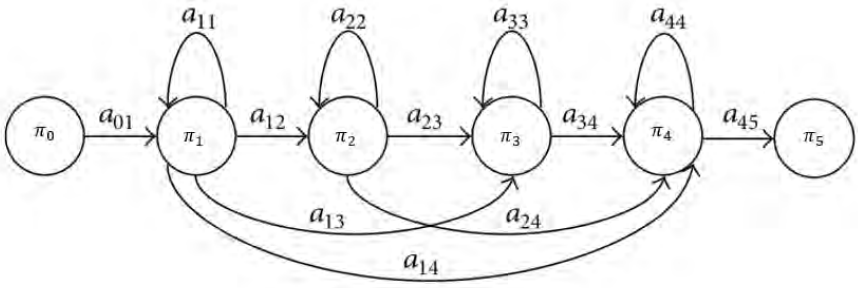


Figure 0-7: HMM Bakis General Left-to-Right Structure

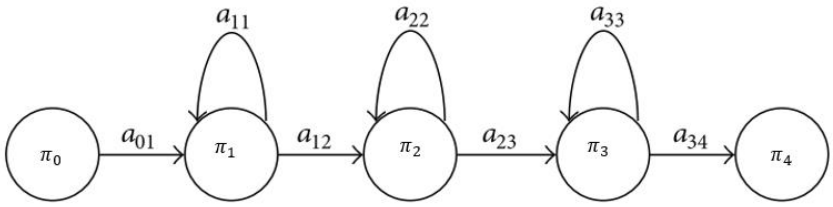


Figure 0-8: HMM Bakis Linear Structure

When the HMM classifier is given a new observation, it provides the distribution of probability among classes. The principle used is derived from generative probability models [71] such as a Naïve Bayes Classifier (NB). It calculates the subsequent probability of class using the distribution of input features. The possibility of a feature set being fit to be classified, as a particular label, is calculated using the Bayes theorem. Given a sequence $O = o_1, o_2, \ldots o_T$, this is mathematically represented by equations 4.4 and 4.5:

$$chose\_class = arg\max_{class}\left[P(M_{class}|O)\right] \tag{4.4}$$

By Bayes rule:

$$P(M_{class}|O) = \frac{P(class) \times P(O|M_{class})}{P(O)}, \tag{4.5}$$

where $P(class)$ is the prior likelihood of a label, $P(O|M_{class})$ is the previous likelihood that the given features can be classified as the label and $P(O)$ is the possibility that the previous

class has the given feature. Probabilistic classifiers, especially the NB, have shown to be successful in experiments with an insufficient amount of data. However, this classifier make vastly simplified probabilistic assumptions [71]

This model has been used in many variations in ASR research. Guatam and Soni [72] evaluated the efficiency of HMM in perpetual talk affirmation, using the existing Defence Advanced Research Projects Agency (DARPA) database. In this work, the authors used HMM language structure to overcome syntax and semantics constraints. A summary of ASR research employing HMM is tabulated in Table 4.2.

Table 0.2: Summary of Automatic Speech Recognition Research That Specifically Uses Hidden Markov Model For Classification

Classification technique	**Type of Speech**	**Reference**	**Number of Subjects**	**Hours of voice**	**Vocabulary Size**	**Word Classification Accuracy**
5 state HMM	Isolated	[45]	Not specified	Not specified	5	87.6%
5 state Continuous Density Hidden Markov Model	Continuous	[28]	4	1.3	4	82.5%
3 state HMM (unsupervised acoustic modelling)	Mixed	[73]	210	56.23	25 651	34%

6 state HMM	Continuous (language modelling)	[72]	113	2.75	not specified	99%
5 state HMM	Isolated	[74]	4	2 400	10 (numbers)	98%

HMM are popular for their left-to-right structure and the ability to treat each sound separately. Despite its appraisal, HMMs are computationally and memory expensive [71]. For an n length sequence, requiring the most likely transition between s states and e edges. The memory is proportional to sn and computational time is proportional to en. This makes the HMM very slow for real-time speech-to-text conversion.

1.20.3 Deep Neural Network

DNN are a subfield of ANN, which is inspired by the human brain neuron connections [55]. Neurons are connected in parallel to form a network. The networks function is determined by the connection between the neurons. The network is trained so that a certain input has a defined output. In DNN there are multiple hidden layers of neurons, which receive inputs from external stimuli and other neurons.

DNN architecture may be differentiated according to the neuron layer structure, i.e., fully connected, convolutional, and recurrent layers. Fully connected layers are also known as dense layers or multi-layer perception (MLP). MLP has been successful in extracting higher-level

representations of ASR [75]. Convolutional neural networks (CNN) use a small size 2D convolution kernel sweep. In image processing, the convolutional layer extract features and preserves a spatial relationship between the input pixels [76]. This has encouraged the use of CNN in ASR as the spectral representation of the speech signal can be treated as an image. Lastly, RNN allows cyclical connection. These connections enable the network to access previously processed inputs, introducing the vanishing gradient problem when training. To compensate for this shortcoming, two units, termed long short-term memory (LSTM) and gated recurrent unit (GRU), were introduced by Hochreiter and Schmidhuber [77] and Cho *et al.* [78].

In addition to the layer structure, DNNs can be applied in different ASR approaches i.e. front-end, back-end, and joint technique [79]. Front-end technique is mainly for feature enhancement, back-end is for language modelling, and the joint technique combines front- and back-end techniques. DNNs have become popular in ASR as techniques that address the noise robustness problem. The network has also been used for multilingual DNN cross-language transfer[80], which allows the sharing of hidden layers for several under-resourced languages. The multitask learning (MTL) model proposes exploiting universal phonemes in a multi-lingual ASR, where the system will only require language-specific triphones.

In its application in the back-end technique, DNNs use the discriminative features, are efficient, and deal well with non-stationary noise, if presented with sufficient training data. On the other side, most DNNs are highly dependent on HMMs, and require a relatively large amount of data. Hence the application of cross-language transfer for under-resourced languages was proposed [81]. A summary of DNN-based ASR research is tabulated in Table 4.3.

Table 0.3: Summary Of Automatic Speech Recognition Research That Specifically Uses Deep Neural Network For Classification

Classification technique	**Type of Speech**	**Reference**	**Number of Subjects**	**Vocabulary Size**	**Word Classification Accuracy**
RNN (end-to-end)	Continuous	[81]	284	20 000	88.5%
Multilingual DNN	Mixed	[82]	626	11 269	45%
Context Dependent CD-DNN-HMM	continuous	[75]	24	65 000	69.6%
Multilingual DNN (3 languages)	Isolated	[80]	Afrikaans– 160 seSotho – 162 siSwati - 156	1 159 1 513 1 833	7.3% (WER) 19.0% (WER) 16.8% (WER)

CNN	Bigram language model	[46]	462	11	33.4% (WER)

1.21 Recognition Theory

In this research, DNNs are proposed for word classification. This choice was encouraged by the growing interest in DNNs for multi-lingual ASR. While ANN algorithms behave much like the human brain, capable of self-learning, they can also be trained under supervision.

The building block for ANN is called perceptron, which is an algorithm inspired by biological neurons. The human brain has approximately 100 billion neurons [58]. A small biological neural network is illustrated in Fig. 4-9. The dendrites receive inputs from other neurons or external stimuli. The input is processed by the axon. Near the end of the neuron cell, the axon branches into several endings called axon terminals. All the input data is added up in the axon, if the sum is greater than a certain threshold, the axon terminal will be fired up and an output is produced. The output is transmitted to other neurons in a space called the synapses.

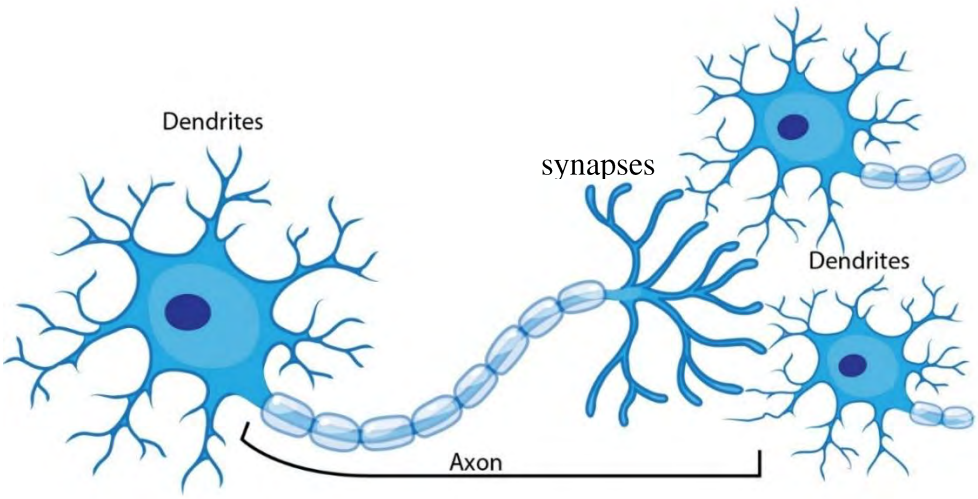


Figure 0-9: Small Biological Neuron Network. Image Taken From https://askabiologist.asu.edu/plosable/speed-human-brain

An ANN perceptron is a mathematical model that loosely captures the essence of the biological neuron functionality. A simple perceptron is illustrated in Fig. 4-10. The perceptron receives multiple data from other perceptrons or external stimuli. The weighing can either amplify or compress the input before it is summed up and transferred to the activation function. If the sum is greater than a certain threshold the activation function will generate an output transferred to other artificial neurons, or another prediction is required. There is a wide variety of activation functions, the most common in ANN are listed in Fig. 4-11.

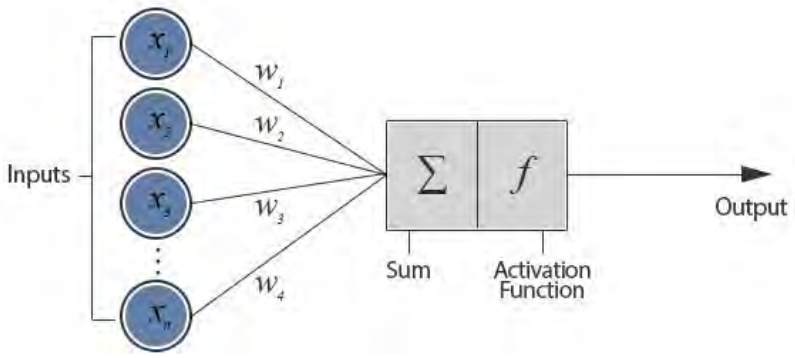


Figure 0-10:Simple Feed-Forward Perceptron. Image Taken from https://www.cc.gatech.edu/~san37/post/dlhc-fnn/

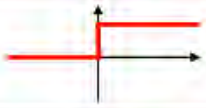
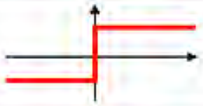
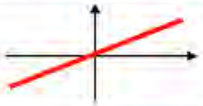
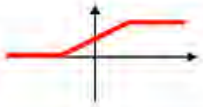
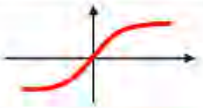
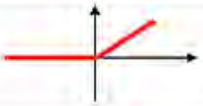
Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

Figure 0-11:Examples of Activation Functions. Image from https://sebastianraschka.com/

There are two types of ANNs available, shallow networks and deep networks. This study focuses on the latter. Under the shallow network umbrella, there is an algorithm called the MLP. MLPs are considered shallow networks if they consist of one layer and are considered DNN if layers are more than one, and each perceptron is connected to every perceptron of the next layer; this is illustrated in Fig. 4-12. MLP is the most traditional DNN architecture. Other

popular DNN architectures are CN), and RNN; these are shown in Figs. 4-13 and 4-14, respectively.

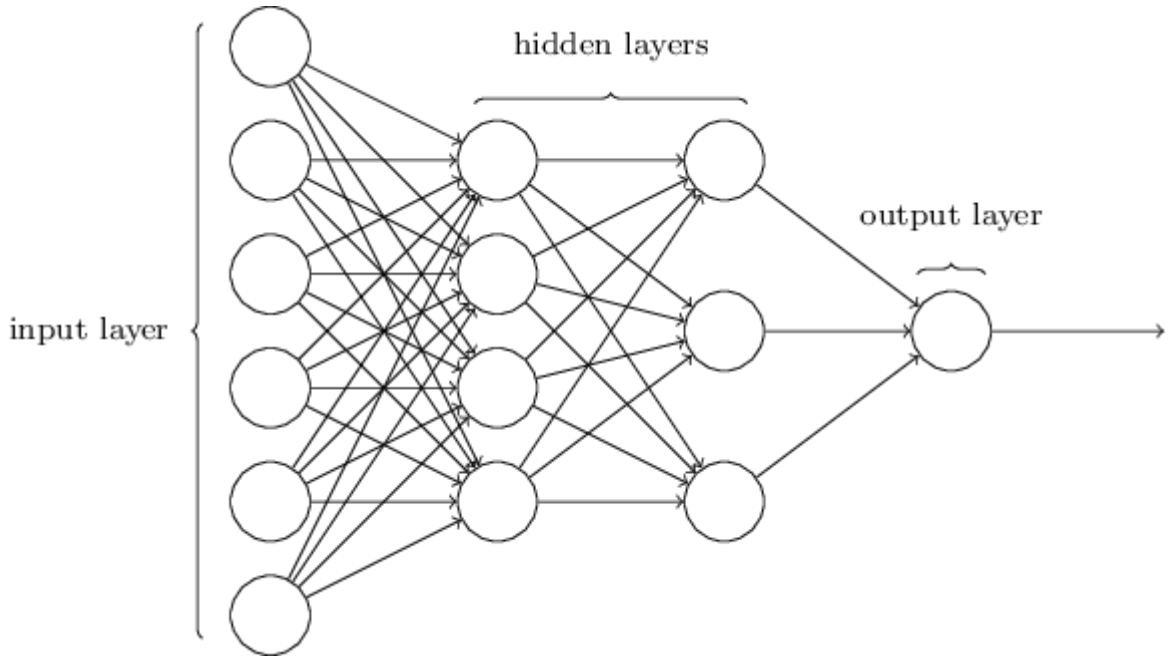


Figure 0-12:Multilayer Perceptron Structure

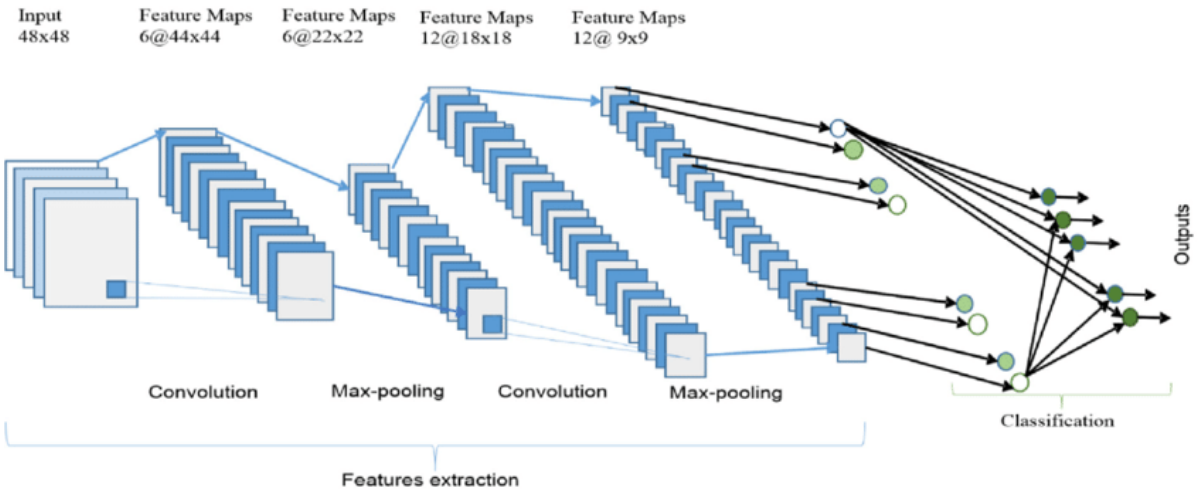


Figure 0-13:Convolution Neural Network Structure [74]

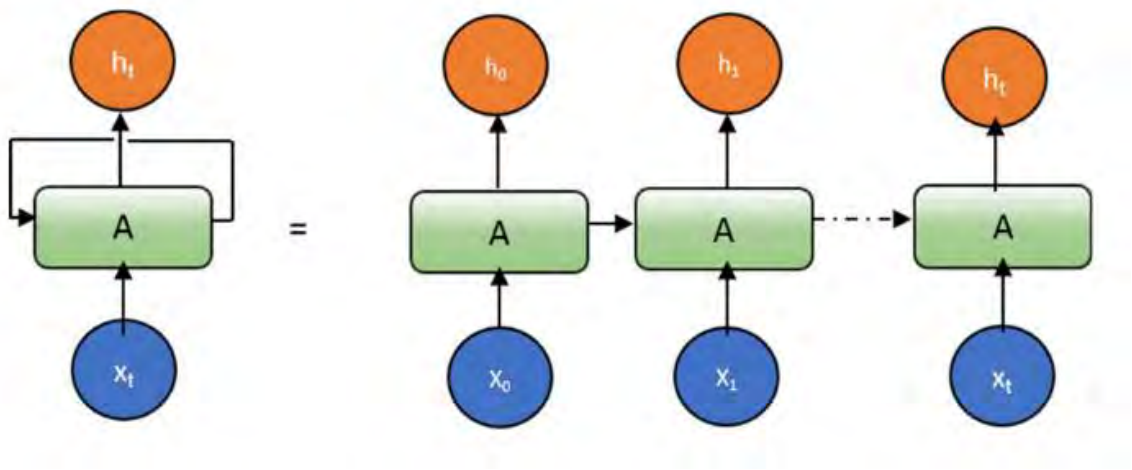


Figure 0-14:Recurrent Neural Network Structure [74]

For the most part, MLPs are limited to classification and regression. They receive tabulated data and transmit the data in a feed-forward manner with weights throughout the whole network. Initially, weights are assigned randomly, thereafter the network applies a repeated process to manage the weight until the error slope is zero.

A CNN has two main stages, including feature extraction and classification; this is shown in Fig. 4-13. CNNs are popular for their ability to process data like the human visual processing system, making the processing of 2D and 3D structures highly optimised. With sparse connections, CNNs have fewer parameters than their fully connected counterparts.

RNNs process data like the human thought processing system, allowing operation over a sequence of vectors over time. RNNs are popular for being able to allow sequence in the input, output, or both. Due to the loop effect, RNNs are infamous for greater gradient vanishing. However, when it is used alongside LSTM [77] or GRU [78], RNNs become powerful tools in

language modelling. A comparative summary of DNN architectures is tabulated in Table 4.4. For time sequence application RNN is more favourable.

Table 0.4: Comparison of DNN Structures Summary [83]

	MLP	**RNN**	**CNN**
Input Data	Tabular	Sequence	Image
Recurrent connection	✗	✓	✗
Parameter Sharing	✗	✓	✓
Spatial relationship	✗	✗	✓
Vanishing or exploding gradient	✓	✓	✓

1.22 Training and validation

For this research 10 isiZulu words were used. These words were extracted from the sentences in the NCHLT database. For training the recognition algorithm, the words were manually extracted from the sentences. RNN-LTSM was trained using a different number of hidden layers. This was done to compare the effect of the number of hidden layers on recognition accuracy.

1.22.1 Dataset

Ten words in Table 4-5 were used to train the RNN-LSTM network. These words were manually extracted from sentences using an audio labeller application to map the ground truths. For the classification to reflect the true effectiveness of isiZulu word recognition, the words had to have syllables mostly used in the Nguni language family. The number of repetitions in the NCHLT database was considered as well. Each word must have at least 200 repetitions and an even distribution between male and female. Though some words were repeated by the same voice the sentences were different. To avoid overfitting the data was split into 60% training data which was used to fit the model and 40% testing data to evaluate the model. The database information is tabulated in Table 4.5.

Table 0.5: Isolated Word Database

Word	Number of Samples	Female	Male	Training data	Testing data
Imali	199	99	100	119	80
lomkhandlu	200	98	102	120	80
lokuqala	203	93	105	122	81
Ngendlela	200	100	100	120	80
Ulwazi	200	101	99	120	80
Abantu	206	106	100	124	82
Ehhovisi	201	93	108	121	80
Isitifikedi	199	99	100	119	80
Ngokulandela	203	96	107	122	81

Ukusebenzisa	200	100	100	120	80

1.22.2 RNN-LSTM network training

The network architecture consisting of 5 layers is illustrated in Fig. 4-15. The audio features are fed to the network in the sequence input layer. The LSTM layer learns the long-short-term dependency of the input sequence. The fully connected layer multiplies the input by a weight matrix and adds a bias vector; all neurons from the previous layer are connected to all neurons in the current layer. The softmax layer provides the output unit activation function mathematically represented by equation 4.6 [84].

Figure 0-15: RNN-LSTM Architecture Block Diagram

$$P(c_r|x,\theta) = \frac{e^{(a_r(x,\theta))}}{\sum_{j=1}^{k} e^{(a_j(x,\theta))}}, \tag{4.6}$$

where k is the number of hidden units, $0 \leq P(c_r|x,\theta) \leq 1$ and $\sum_{j=1}^{k} P\left(c_j\middle|x,\theta\right) = 1$. The last layer takes the softmax function and assigns it to K classes using the cross-entropy function. For updating cells and hidden states the *tanh* activation function was applied and sigmoid was used for gates activation. The number of hidden layers were varied to find the most favourable

number, based on training time, validation accuracy, and the recall accuracy of the test words. Recall accuracy is the ratio of the number of correct predictions over the total number of predictions.

1.22.3 RNN-LSTM Training Options

Standard gradient descent (SGD) is an algorithm that updates the network's weights and biases to minimise the loss function. This is achieved by taking small steps towards negative gradient loss at each iteration. The features are inputted in the network using mini batches, i.e. smaller cuts of the signal. Using equation 4.7 to determine the next iteration's parameter vector, the algorithm evaluates the parameter vector (θ_l) for iteration (l), calculates loss function ($E(\theta_l)$), multiplies the gradient of the loss function ($\nabla E(\theta_l)$) by learning rate ($\alpha > 0$).

$$\theta_{l+1} = \theta_l - \alpha \nabla E(\theta_l) \tag{4.7}$$

The size of mini-batches was chosen based on the size of the computer's processor. To avoid processor overload mini-batches were sized to 27 samples per batch. The full pass of the training algorithm through all mini-batches is called one epoch. The number of epochs was kept low at 30 epochs, to avoid overfitting. Iterations are the number of mini-batches multiplied by the number of epochs. For this research there was a maximum of 1 320 iterations.

The RNN-LSTM network was trained using the algorithm from the MATLAB deep learning toolbox. There are several SGD algorithms, including, stochastic gradient descent with momentum (SGDM), root mean square propagation (RMSProp), and adaptive moment estimation (Adam). This research employed the Adam optimiser, which is a combination of the former two. The Adam optimiser keeps an element-wise moving average for the parameter gradients (m_l) given by equation 4.8 and their square values (v_l) given by equation 4.9, with

gradient decay factors β_1 and β_2, respectively [85]. These moving averages are then used to update the network parameters as expressed in equation 4.10, where ϵ is a small constant used to avoid division by zero.

$$m_l = \beta_1 m_{l-1} + (1 - \beta_1)\nabla E(\theta_l) \tag{4.8}$$

$$\upsilon_l = \beta_2 \upsilon_{l-1} + (1 - \beta_2)[\nabla E(\theta_l)]^2 \tag{4.9}$$

$$\theta_{l+1} = \theta_l - \frac{\alpha m_l}{\bar{\upsilon}_l - \epsilon} \tag{4.10}$$

The RNN-LSTM network was trained using MATLAB. The stochastic gradient descend algorithm was used to optimise the RNN-LSTM algorithm. The training optimization of the implementation of the training option above is illustrated in Fig. 4-16; the number of hidden layers was 250. These results show that the maximum 30th epoch the validation accuracy was 73.19%, the loss was 1 and the training time was 17 min 9 seconds.

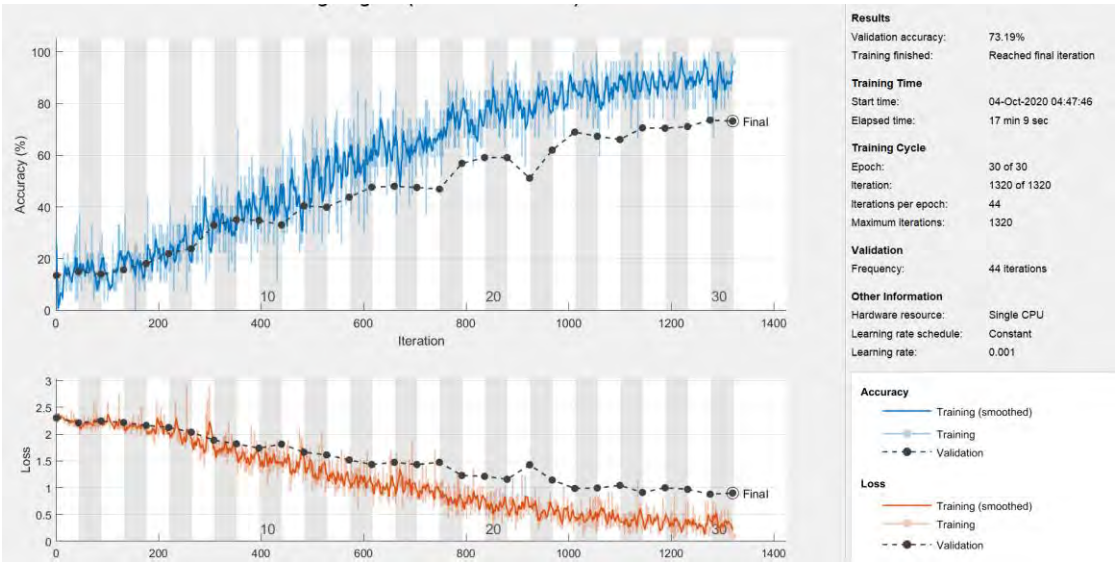


Figure 0-16: RNN-LSTM Network Training Monitor For 250 Hidden Layers

1.23 Results and Discussion

Hidden layers were varied from 100 hidden layers to 400 hidden layers. Where the network used 400 hidden layers, the validation accuracy dropped significantly; compared to 300 hidden layers, the training time also increased by 10 minutes. Increasing the number of hidden layers would not have improved the results. The comparison of varying hidden layers is summarised in Table 4.6. Individual word recall accuracy was only conducted on the first four networks; the results are tabulated in Table 4.7 and graphically illustrated in Fig. 4-17.

Table 0.6: Number of Hidden Layers Comparison Summary

	100 hidden layers	**200 hidden layers**	**250 hidden layers**	**300 hidden layers**	**400 hidden layers**
Training time	9 min 8 sec	20 min	18 min 20 sec	25 min 6 sec	34 min 56 sec
Validation accuracy	61.3%	65.83%	73.07%	79.62%	68.42%
recall accuracy	63.8%	64.24%	73.41%	78.18%	64.61%

Table 0.7: Isolated Word Recall Accuracy Base on The Number Of Hidden Layers

Word	100 hidden layers	200 hidden layers	250 hidden layers	300 hidden layers
Abantu	50%	55%	88%	90%
Ehhovisi	68%	61%	78%	81%
Imali	73%	63%	63%	73%
Isitifikedi	81%	58%	89%	54%
lokuqala	71%	64%	77%	86%
lomkhandlu	35%	54%	59%	44%
Ngendlela	59%	93%	89%	90%
Ngokulandela	89%	48%	44%	75%
Ukusebenzisa	75%	78%	84%	84%
Ulwazi	39%	69%	64%	73%

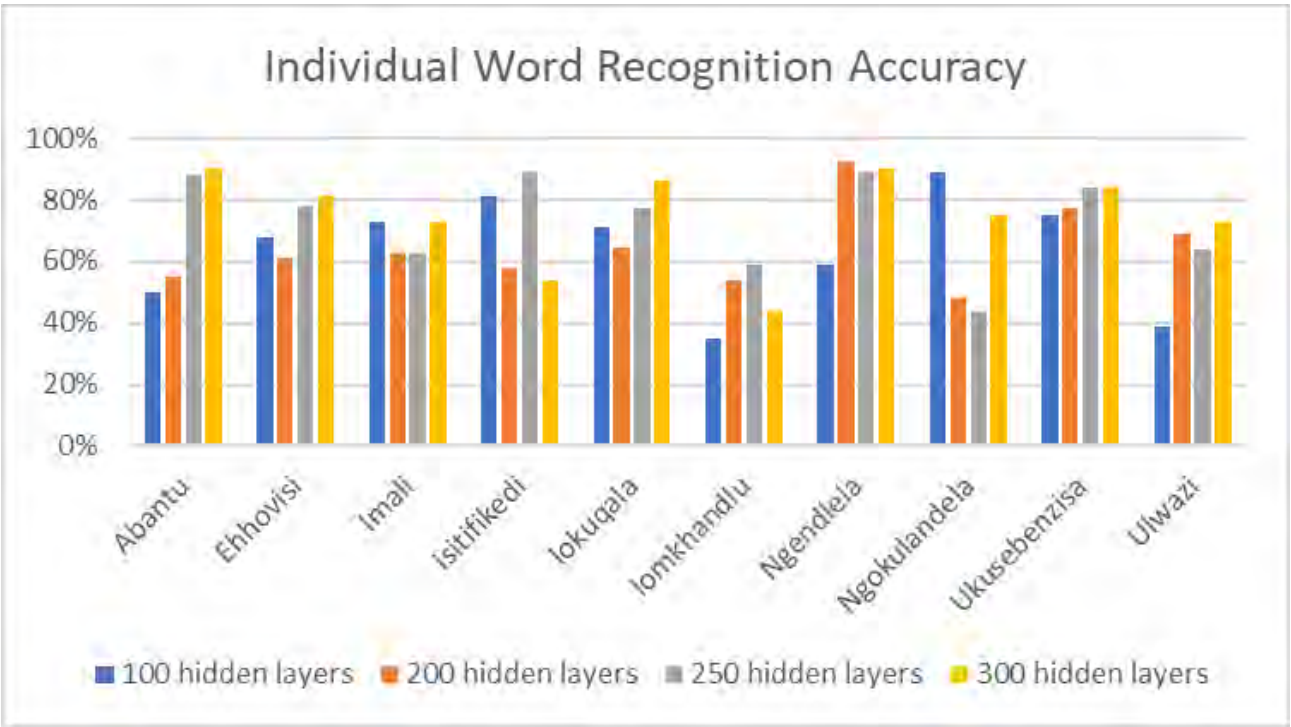


Figure 0-17: Word Recognition Accuracy Comparison

The results of each word recall accuracy in the different networks are tabulated in Table 2.7 and represented graphically in Fig. 4-17. These results indicate that the words recall accuracy for “isitifiketi” and “lomkhandlu” deteriorated as the number of layers increased. In contrast, the recall accuracy for the other eight words improved as the number of layers increased. Therefore, the network with 300 hidden layers was chosen. The resultant confusion matrix for this algorithm is shown in Fig. 4-18 and interpreted in table 4.8.

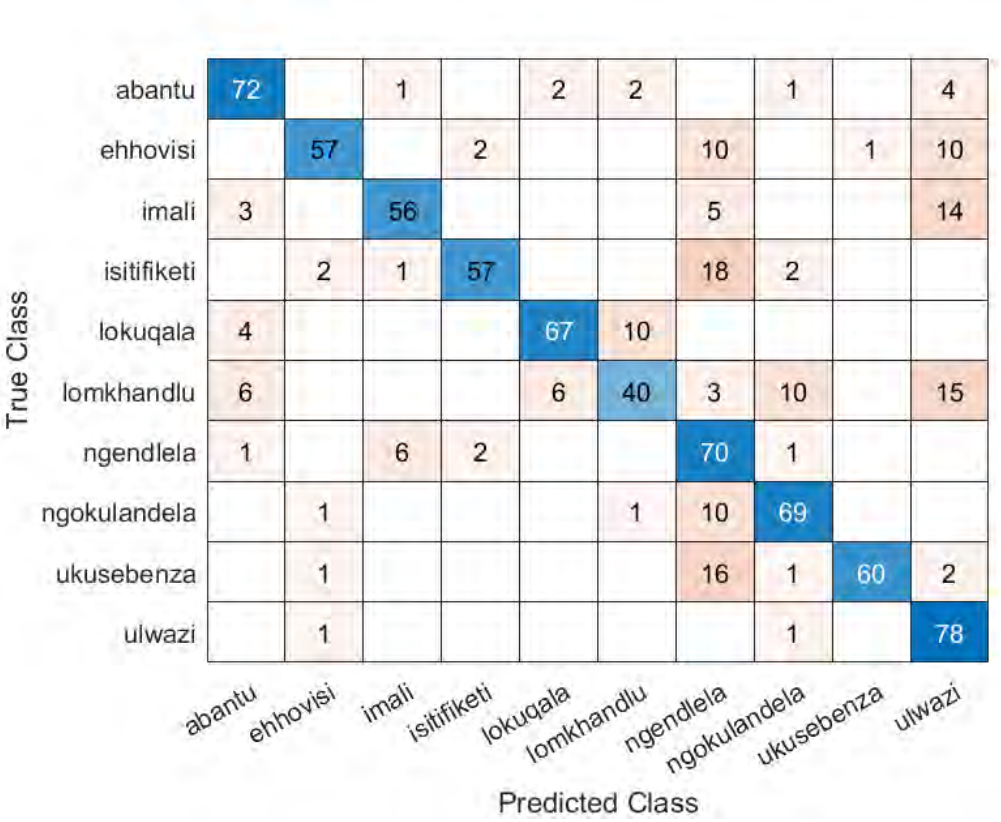


Figure 0-18: Isolated Word Confusion Matrix

Table 0.8: RNN-LSTM Recall and Precision Rates for Isolated Words

Word	**Recall Rate**	**Precision rate**
Abantu	83.7%	87.8%
Ehhovisi	91.9%	71.3%
Imali	87.5%	71.8%
Isitifikedi	93.4%	71.3%

lokuqala	89.3%	82.7%
lomkhandlu	75.5%	50.0%
Ngendlela	53.0%	87.5%
Ngokulandela	81.2%	85.2%
Ukusebenzisa	98.4%	75.0%
Ulwazi	63.0%	97.5%

As aforementioned in chapter 3, the uneven distribution of speaker recordings affected the recall and precision rate. While on par with chapter 3 results, it can be noted that the accuracy significantly improved by using RNN-LSTM. The uneven distribution made the recognition system more speaker-dependent.

1.24 Conclusion

A survey of the ASR classification literature highlighted 3 dominant classification methods; these include SVM, HMM, and DNN. The latter was proposed considering its dominance in recent ASR research for under-resourced languages. Three DNN architecture models, viz. MLP, CNN, and RNN, were introduced and compared to find the most favourable for this research. For its ability to process sequential information RNN was proposed for implementation. Five RNN networks were implemented and compare based on training time,

validation, and recall accuracy. The RNN-LSTM with 300 hidden layers was elected. It achieved an average validation accuracy of 79.62% and 78.18% recall accuracy. The individual word recall accuracy ranged from 44% to 90%, and only 20% of the database scored below 70% recall. These results are encouraging considering data limitations.

ALGORITHMS EVALUATION

1.25 Introduction

Three ASR building blocks have been independently discussed namely, automated word boundary detection in chapter 2, MFCC feature extraction in chapter 3 and RNN classifier in chapter 4. This chapter combines these blocks to compare the different algorithms. The NCHLT database has recordings in sentence structures averaging 3 words per sentence. The classification algorithm discussed in chapter 4 is based on isolated words. This research attempts to find a solution for real-life isiZulu ASR. This chapter focuses on the use of two algorithms:

1. In-sentence word recognition without explicit boundaries which is illustrated in Fig. 5- 1. In this algorithm the speech signal features are extracted without explicit word boundary
2. In-sentence word recognition with boundary estimation which is illustrated in Fig. 5-2. In this algorithm the speech signal is processed to delineate word boundaries before feature extraction.

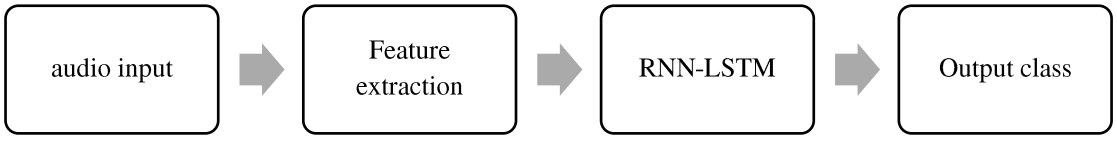


Figure 0-1: In-Sentence Word Detection and Classification Block Diagram

Figure 0-2: Word Boundary Estimation Before Classification Block Diagram

1.26 Simulation Set-up

Table 0.1: Algorithm Test Dataset

Sentences	Word in RNN_LSTM training dataset	Number of samples		
		female	male	Grand Total
abantu abafuna isondlo	Abantu	22	19	41
ehhovisi eliseduzane labezindaba	Ehhovisi	18	15	33
imali efanele okumele	Imali	18	20	38
imibhalo ezethula ulwazi	Ulwazi	22	23	45
intela ngendlela evikelekile	Ngendlela	15	16	31
iphepha lokuqala p	Lokuqala	15	17	32
isitifikedi sakho sokuzalwa	Isitifikedi	15	18	33
lomkhandlu ophethe kumele	Lomkhandlu	18	19	37

ngokulandela umyalelo wesondlo	Ngokulandela	17	16	33
nokucabangisisa ukusebenzisa ulimi	Ukusebenzisa	14	18	32
Grand Total		**174**	**181**	**355**

For each of the ASR stages discussed, datasets were structured to suit the requirements for each stage. For word boundary estimation, whole sentences were used, and for speech recognition, words were completely isolated. The final test data was sorted so there would be no bias in the final recognition. All recordings in the sentence segmentation training and testing data were part of the speech recognition data. To account for all words used to train the neural network, 10 different sentences containing the RNN-LSTM network training dataset were selected. The dataset used for this simulation is tabulated in Table 5.1.

1.27 Results and Discussion

For the simulation performed, recognition accuracy was the only performance parameter considered. The accuracy was calculated as a fraction of the sum of all correct predictions i.e. TP +TN over all predictions as presented in equation 5.1. The addition of the boundary estimation stage increased the processing time. Comparative results are tabulated in Table 5.2. In-sentence detection refers to the recognition of a word in a complete sentence without segmentation. Word boundary estimated recognition accuracy refers to words that were correctly recognised in their correct position. A detailed recognition with segmentation performance is tabulated in Table 5.3. True positives refer to words recognised in the correct

sentence and position, and false positives refer to words detected in the correct sentence but incorrect position.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Table 0.2: Word Recognition Algorithm Accuracy Comparison

Word	**Isolated Word**	**In-sentence Detection (bulk processing)**	**In-sentence Detection (individual processing)**	**Word boundary estimated (individual processing)**
Abantu	90%	51.22%	0%	9.75%
Ehhovisi	81%	0%	0%	0%
Imali	73%	8.77%	0%	0%
Isitifikedi	54%	0%	0%	63.64%
lokuqala	86%	6.25%	43.75%	46.88%
lomkhandlu	44%	2.70%	10.81%	27.03%
Ngendlela	90%	48.39%	0%	0%
Ngokulandela	75%	7.07%	51.52%	63.64%

Ukusebenzisa	84%	0%	0%	21.88%
Ulwazi	73%	2.22%	62.22%	0%
Average Accuracy	**78.18%**	**15.68%**	**17.75%**	**23.28%**

Table 0.3: Detailed Word Recognition Accuracy with Word Boundary Estimation

Sentences	**Word Boundary Estimation dataset**	**Word in RNN_LSTM training dataset**	**Accuracy**		
			Sentence word count	**True Positives**	**False-positive**
abantu abafuna isondlo	✓	Abantu	✓	9.76%	2.44%
ehhovisi eliseduzane labezindaba	✓	Ehhovisi	✓	0%	0%
imali efanele okumele	×	Imali	×	0%	0%
imibhalo ezethula ulwazi	×	Ulwazi	×	0%	0%
intela ngendlela evikelekile	✓	Ngendlela	✓	0%	0%
iphepha lokuqala p	✓	Lokuqala	×	46.88%	43.75%

isitifikedi sakho sokuzalwa	✓	Isitifikedi	✓	63.64%	36.36%
lomkhandlu ophethe kumele	×	Lomkhandlu	✓	27.02%	13.51%
ngokulandela umyalelo wesondlo	×	Ngokulandela	✓	63.64%	0%
nokucabangisisa ukusebenzisa ulimi	×	Ukusebenzisa	✓	21.89%	21.89%

Several factors apart from the recognition algorithm affect accuracy. The recognition accuracy of isolated words is higher due to the type of training dataset. The RNN-LSTM was trained using manually isolated words; hence, the classifier is biased to recordings that are standalone voice signals.

Language structure is an important aspect of ASR. However, this research is limited to signal processing of isiZulu speech. When the RNN-LSTM network was given full sentences to determine if the network can detect a word in that signal, “ulwazi” achieved the highest accuracy.

Considering that only five of the sentences in this database were used to train the RUSBoost word boundary estimator, these sentences were expected to achieve high recognition. However, this is not the case; in this simulation, two words that were on the RUSBoost training data, “ehhovisi” and “ngendlela”, were classed as out of vocabulary words. Meanwhile, “ngokulandela” had the highest recognition accuracy, but not in the RUSBoost training data. It was also the second-highest in the in-sentence detection algorithm.

Table 5.4 presents the correlation between recognition accuracy, word size, and position. It can be noted that words with a higher number of frames have better accuracy when word

boundaries have been estimated. This is because more frames provide a better feature representation. The number of frames makes it less likely for the whole word to be misclassified as a boundary region. While the length of the word does improve recognition, it should be noted that the amount of unvoiced phonemes in the word lowers the accuracy for boundary detection, thus lowering recognition accuracy. It can also be noted that the position of the word has little to no effect on the accuracy after boundary estimation. The word "lokuqala" is the only word with a click sound, its uniqueness makes it easier to recognise regardless of the number of frames or position.

Table 0.4: Word Recognition Accuracy and Word Size Correlation

Word	**Average of frames**	**word position**	**In-sentence Detection (individual processing)**	**Word boundary estimated (individual processing)**
Isitifikedi	117	1	0%	63.64%
Ukusebenzisa	115	2	0%	21.88%
Ngokulandela	112	1	51.52%	63.64%
lomkhandlu	105	1	10.81%	27.03%
lokuqala	101	2	43.75%	46.88%
Ehhovisi	92	1	0%	0%
Ngendlela	83	2	0%	0%
Ulwazi	80	3	62.22%	0%
Abantu	65	1	0%	9.75%
Imali	64	1	0%	0%

CONCLUSION

1.28 Word Boundary Estimation

Word co-articulation in isiZulu makes a binary Gaussian probability algorithm ineffective for voice activity detection. Consequently, prosodic features and HOS were proposed in addition to a binary Gaussian probability algorithm. To further optimise the use of word boundary estimation in ASR, the problem was converted to a classification problem. However, only 9% of the training data had boundary frames, and this necessitated a classifier that will accommodate this imbalanced data. RUSBoost was proposed for classification.

The results considered the automation of boundary recognition. The use of rudimentary and HOS features, with recognition automated, using the RUSBoot classifier, produced encouraging results, with boundary region recognition of 68.2%. The dataset used was diversified such that the approach was not biased. For each of the 5 sentences used, individual confusion matrices were plotted, and the boundary recognition ranged from 51.5% to 77.3%.

1.29 Feature Extraction

Several feature extraction methods have been proposed in the past. The literature review suggested two dominant methods viz. LPC and MFCC. The latter was identified as the favourable method based on feature extraction work that has been in the field. After experimenting with 9 to 17 coefficients using *k*-NN as a classifier and a 5-word database, 13 coefficients achieved the best results at 63.14%. Each coefficient contains cepstral, change in velocity, and acceleration, bringing the number of coefficients to 39.

1.30 Recognition

A survey of the ASR classification literature highlighted 3 dominant classification methods viz. SVM, HMM, and DNN. These methods were exhaustively discussed. Recent literature highlights DNN as the dominating classifier for under-resourced languages, as the research community is exploring the development of multi-lingual ASR [86]. For this reason, DNN was proposed.

All three DNN architecture models i.e., RNN, CNN, and MLP, were discussed at length and compared to find the most favourable for this research. RNN proved to be most favourable compared to CNN and MLP. RNN is more susceptible to vanishing gradient compared to the CNN and MLP. To mitigate this shortfall, LSTM was employed. Five RNN-LSTM algorithms were implemented and compared, based on training time, validation, and recall accuracy. The RNN-LSTM with 300 hidden layers was elected, achieving 79.6% validation accuracy and 78.18% recall accuracy. Individual word recall accuracy ranging from 44% to 90% with only 20% of the database scoring below 70% recall. These results are encouraging considering data limitation.

1.31 Algorithm comparison

Despite the encouraging accuracy achieved in isolated word recognition, words in real-life spoken sentences are hardly completely isolated. Therefore, an algorithm that can detect and recognise words in a sentence was required. The trained neural network was tested to determine

if it will be able to detect and correctly classify the word in a sentence, without explicit boundaries as well as the effect of boundary estimation on the overall recognition.

The results were not too far apart, detection without explicit boundaries achieved 17.75% and recognition with demarcated boundaries achieved 23.28%. The difference between the two algorithms is the factors affecting accuracy. Where there were no explicit boundaries, the major factor affecting accuracy was the position of the word in a sentence. Words in first and last positions had greater accuracy, whereas with explicit boundaries the major factor is the duration of the word. This is justifiable, as classifiers with imbalanced data tend to favour classes with more samples.

The boundary estimation algorithm proposed achieved an overall 70% word count accuracy with 52% of words recognised in the correct position. Results can be improved by increasing training data. This can be achieved by using closely related languages to develop a multi-lingual ASR system.

1.32 Further research

Further investigation of word boundary features such as adaptive time frequency parameters is proposed. This may assist in reducing the number of features required thus reducing computing cost. The global trend for word recognition in under-resourced languages is the used of DNN and closely related languages. It is proposed that Nguni languages be used with feature extraction inclusion in the DNN structure.

REFERENCES

[1] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication,* vol. 56, pp. 85–100, 01/31 2014, doi: 10.1016/j.specom.2013.07.008.

[2] E. W. D. Whittaker, "Statistical language modelling for automatic speech recognition of Russian and English," Degree of Doctor of Philosophy, Cambridge University Engineering Departmen, Trinity College, University of Cambridge, London, 2000.

[3] S. Krauwer, "The basic language resource kit (BLARK) as the first milestone for the language resources roadmap," in *SPECOM*, Moscow, 2003: ELSNET.

[4] B.-H. Juang and L. R. Rabiner, "Automatic speech recognition–a brief history of the technology development," *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara,* vol. 1, p. 67, 2005.

[5] A. L. Guzman, "Voices in and of the machine: Source orientation toward mobile virtual assistants," *Computers in Human Behavior,* vol. 90, pp. 343-350, 2019/01/01/ 2019, doi: https://doi.org/10.1016/j.chb.2018.08.009.

[6] L. Deng and X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 5, pp. 1060-1089, 2013, doi: 10.1109/TASL.2013.2244083.

[7] M. Alexander, "Edging Down The Road To Voice Recognition," (in English), *Computerworld,* vol. 24, no. 3, p. 17, 1990 Jan 15

2017-11-14 1990. [Online]. Available: https://search.proquest.com/docview/215979909?accountid=10612.

[8] B. Juang and L. Rabiner, "Automatic Speech Recognition - A Brief History of the Technology Development," 01/01 2005.

[9] H. Olson and H. Belar, "Phonetic typewriter," *IRE Transactions on Audio,* vol. AU-5, no. 4, pp. 90-95, 1957, doi: 10.1109/TAU.1957.1166018.

[10] P. Denes, "The design and operation of the mechanical speech recognizer at University College London," *Journal of the British Institution of Radio Engineers,* vol. 19, no. 4, pp. 219-229, 1959.

[11] S. Furui, "Fifty years of progress in speech and speaker recognition," *Journal of The Acoustical Society of America - J ACOUST SOC AMER,* vol. 116, pp. 2497-2498, 10/01 2004, doi: 10.1121/1.4784967.

[12] D. Reddy, "An approach to computer speech recognition by direct analysis of the speech wave (Tech. Report No. C549)," ed: Stanford, CA: Stanford University, Computer Science Department, 1966.

[13] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 23, no. 1, pp. 67-72, 1975, doi: 10.1109/TASSP.1975.1162641.

[14] L. Rabiner, S. Levinson, A. Rosenberg, and J. Wilpon, "Speaker-independent recognition of isolated words using clustering techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 27, no. 4, pp. 336-349, 1979, doi: 10.1109/TASSP.1979.1163259.

[15] J. Bing-Hwang and S. Furui, "Automatic recognition and understanding of spoken language - a first step toward natural human-machine communication," *Proceedings of the IEEE,* vol. 88, no. 8, pp. 1142-1165, 2000, doi: 10.1109/5.880077.

[16] Eberhard, D. M., F. S. Gary, and D. F. Charles. "Ethnologue: Languages of the World. Twenty-third edition." SIL International. (accessed 23 May, 2020).

[17] E. Barnard, M. H. Davel, C. Heerden, F. de Wet, and J. Badenhorst, "The NCHLT speech corpus of the South African languages," presented at the In SLTU-2014, St. Petersburg, Russia, 2014.

[18] I. McLoughlin, *Applied Speech and Audio Processing: With Matlab Examples*. Cambridge University Press, 2009.

[19] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *The Bell System Technical Journal,* vol. 54, no. 2, pp. 297-315, 1975, doi: 10.1002/j.1538-7305.1975.tb02840.x.

[20] A. D. Vijayendra and V. K. Thakar, "Word boundary detection for Gujarati speech recognition using in-ear microphone," in *2016 1st India International Conference on Information Processing (IICIP)*, 12-14 Aug. 2016 2016, pp. 1-6, doi: 10.1109/IICIP.2016.7975324.

[21] S. Jongseo, K. Nam Soo, and S. Wonyong, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters,* vol. 6, no. 1, pp. 1-3, 1999, doi: 10.1109/97.736233.

[22] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2008.

[23] R. M. Dafydd Gibbon, and Richard Winsk, *"EAGLES (Expert Advisory Groups on*

Language Engineering Standards) HANDBOOK of Standards and

Resources for Spoken Language Systems". Berlin & New York,: Walter de Gruyter

Publishers, , 1995.

[24] A. Agarwal, A. Jain, N. Prakash, and S. S. Agrawal, "Word boundary detection in continuous speech based on suprasegmental features for Hindi language," in *2010 2nd International Conference on Signal Processing Systems*, 5-7 July 2010 2010, vol. 2, pp. V2-591-V2-594, doi: 10.1109/ICSPS.2010.5555691.

[25] V. Naganoor, A. K. Jagadish, and K. Chemmangat, "Word boundary estimation for continuous speech using higher order statistical features," in *2016 IEEE Region 10 Conference (TENCON)*, 22-25 Nov. 2016 2016, pp. 966-969, doi: 10.1109/TENCON.2016.7848148.

[26] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: Timit and beyond," *Speech Communication,* Article vol. 9, no. 4, pp. 351-356, 1990, doi: 10.1016/0167-6393(90)90010-7.

[27] A. Al-Sabri, A. Adam, and F. Rosdi, *Automatic Detection of Shadda in Modern Standard Arabic Continuous Speech*. 2018.

[28] P. P. Patil and S. A. Pardeshi, "Marathi connected word speech recognition system," in *2014 First International Conference on Networks & Soft Computing (ICNSC2014)*, 19-20 Aug. 2014 2014, pp. 314-318, doi: 10.1109/CNSC.2014.6906687.

[29] D. Dalva, U. Guz, and H. Gurkan, "Effective semi-supervised learning strategies for automatic sentence segmentation," *Pattern Recognition Letters,* vol. 105, pp. 76-86, 2018/04/01/ 2018, doi: https://doi.org/10.1016/j.patrec.2017.10.010.

[30] Q. Wang, X. Zhao, and J. Xu, "Pitch detection algorithm based on normalized correlation function and central bias function," in *2015 10th International Conference on Communications and Networking in China (ChinaCom)*, 15-17 Aug. 2015 2015, pp. 617-620, doi: 10.1109/CHINACOM.2015.7498011.

[31] S. Gonzalez and M. Brookes, "A Pitch Estimation Filter robust to high levels of noise (PEFAC)," in *2011 19th European Signal Processing Conference*, 29 Aug.-2 Sept. 2011 2011, pp. 451-455.

[32] F. Bahja, E. H. I. Elhaj, and J. D. Martino, "On the use of wavelets and cepstrum excitation for Pitch Determination in real-time," in *2012 International Conference on Multimedia Computing and Systems*, 10-12 May 2012 2012, pp. 150-153, doi: 10.1109/ICMCS.2012.6320184.

[33] D. Hermes, "Measurement of pitch by subharmonic summation," *The Journal of the Acoustical Society of America,* vol. 83, pp. 257-64, 02/01 1988, doi: 10.1121/1.396427.

[34] Mathworks. "Audio Toolbox: User's Guide (R2018a)." The Mathworks Inc. (accessed.

[35] C. Drummond and R. C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling," 2003, 2003.

[36] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans,* vol. 40, no. 1, pp. 185-197, 2010, doi: 10.1109/TSMCA.2009.2029559.

[37] N. Sharma and S. Sardana, "A real time speech to text conversion system using bidirectional Kalman filter in Matlab," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 21-24 Sept. 2016 2016, pp. 2353-2357, doi: 10.1109/ICACCI.2016.7732406.

[38] N. S. Endah, S. Adhy, and Sutikno, "Comparison of Feature Extraction MFCC and LPC in Automatic Speech Recognition for Indonesian," *Telecommunication Computing Electronics and Control,* vol. 15, no. 1, pp. 292-298, 2017.

[39] F. Valente and P. Motlicek, "Detecting and labeling folk literature in spoken cultural heritage archives using structural and prosodic features," *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI),* pp. 1-6, 27-29 June 2012 2012, doi: 10.1109/CBMI.2012.6269839.

[40] D. Dalva, I. D. Revidi, U. Guz, and H. Gurkan, "Extraction and comparison of various prosodic feature sets on sentence segmentation task for Turkish Broadcast News data," *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE),* pp. 70-73, 14-16 May 2014 2014, doi: 10.1109/JCSSE.2014.6841844.

[41] K. Lounnas, L. Demri, L. Falek, and H. Teffahi, "automatic language identification for berber and arabic languages using prosodic features," in *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, 28-31 Oct. 2018 2018, pp. 1-4, doi: 10.1109/CISTEM.2018.8613414.

[42] T. B. Adam, M. S. Salam, and T. S. Gunawan, "Wavelet based Cepstral Coefficients for neural network speech recognition," in *2013 IEEE International Conference on Signal and Image Processing Applications*, 8-10 Oct. 2013 2013, pp. 447-451, doi: 10.1109/ICSIPA.2013.6708048.

[43] A. Shirani and A. R. N. Nilchi, "Speech Emotion Recognition based on SVM as Both Feature Selector and Classifier," *International Journal of Image, Graphics and Signal Processing,* vol. 8, no. 4, pp. 39-45, 2016.

[44] L. M, K. L, and L. A, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters,* vol. 42, pp. 11-24, 2014/06/01/ 2014, doi: https://doi.org/10.1016/j.patrec.2014.01.008.

[45] S. M. Mon and H. M. Tun, "Speech-To-Text Conversion STT System Using Hidden Markov Model HMM," *International Journal of Scientific & Technology Research,* vol. 4, no. 6, pp. 349-352, 2015.

[46] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.,* vol. 22, no. 10, pp. 1533-1545, 2014, doi: 10.1109/taslp.2014.2339736.

[47] K. Dutta and K. K. Sarma, "Dynamic segmentation of vocal extract for Assamese Speech to Text Conversion using RNN," in *2012 2nd National Conference on Computational Intelligence and Signal Processing (CISP)*, 2-3 March 2012 2012, pp. 126-131, doi: 10.1109/NCCISP.2012.6189692.

[48] C. Y. Fook, M. Hariharan, S. Yaacob, and A. Adom, "A review: Malay speech recognition and audio visual speech recognition," in *2012 International Conference on Biomedical Engineering (ICoBE)*, 27-28 Feb. 2012 2012, pp. 479-484, doi: 10.1109/ICoBE.2012.6179063.

[49] P. R. Rao, *Communication Systems*. India: Tata McGraw-Hill Education, 2013.

[50] T. B. Mokgonyane, T. J. Sefara, T. I. Modipa, M. M. Mogale, M. J. Manamela, and P. J. Manamela, "Automatic Speaker Recognition System based on Machine Learning Algorithms," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 28-30 Jan. 2019 2019, pp. 141-146, doi: 10.1109/RoboMech.2019.8704837.

[51] D. Atallah, M. Badawy, A. El-Sayed, and M. Ghoneim, "Predicting kidney transplantation outcome based on hybrid feature selection and KNN classifier," *Multimedia Tools and Applications,* vol. 78, pp. 20383–20407, 07/01 2019, doi: 10.1007/s11042-019-7370-5.

[52] G. M. Venturini, "Statistical distances and probability metrics for multivariate data, ensembles and probability distributions," Ph. D, Department of Statistics, University Carlos III of Madrid, Madrid, Spain, 2015.

[53] P. Domingos and "A few useful things to know about machine learning," *Commun. ACM,* vol. 55, no. Association for Computing Machinery, pp. 78–87, 2012.

[54] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Canada: John Wiley & Son, 2001.

[55] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access,* vol. 7, pp. 19143-19165, 2019, doi: 10.1109/ACCESS.2019.2896880.

[56] R. Saravanan and P. Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 14-15 June 2018 2018, pp. 945-949, doi: 10.1109/ICCONS.2018.8663155.

[57] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Baton Rouge, UNITED STATES: Taylor & Francis Group, 2010.

[58] A. Gupta, " "Introduction to Deep Learning: Part 1"," *Chemical Engineering Progress,* vol. 114, no. 6, pp. 22-29, 2018.

[59] S. Bengio, L. Deng, H. Larochelle, H. Lee, and R. Salakhutdinov, "Guest Editors' Introduction: Special Section on Learning Deep Architectures," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1795-1797, 2013, doi: 10.1109/TPAMI.2013.118.

[60] S. R and V. C. D, "Cross-Entropy Training of DNN Ensemble Acoustic Models for Low-Resource ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 26, no. 11, pp. 1991-2001, 2018, doi: 10.1109/TASLP.2018.2851145.

[61] G. T. Tsenov and V. M. Mladenov, "Speech recognition using neural networks," in *10th Symposium on Neural Network Applications in Electrical Engineering*, 23-25 Sept. 2010 2010, pp. 181-186, doi: 10.1109/NEUREL.2010.5644073.

[62] Y. H. Ghadage and S. D. Shelke, "Speech to text conversion for multilingual languages," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 6-8 April 2016 2016, pp. 0236-0240, doi: 10.1109/ICCSP.2016.7754130.

[63] P. K. Ajmera and R. S. Holambe, "Fractional Fourier transform based features for speaker recognition using support vector machine," *Computers & Electrical Engineering,* vol. 39, no. 2, pp. 550-557, 2013/02/01/ 2013, doi: https://doi.org/10.1016/j.compeleceng.2012.05.011.

[64] G. C. Batista, D. L. Oliveira, O. Saotome, and T. S. Curtinhas, "A New Asynchronous Pipeline Architecture of Support Vector Machine Classifier for ASR System," in *2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 12-14 Aug. 2019 2019, pp. 1-4, doi: 10.1109/INTERCON.2019.8853577.

[65] S. Herrero-Lopez, "Chapter 20 - Multiclass Support Vector Machine," in *GPU Computing Gems Emerald Edition*, W.-m. W. Hwu Ed. Boston: Morgan Kaufmann, 2011, pp. 293-311.

[66] Georgia Tech College of Computing. "Recognition with Bag of Words." https://www.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj4/html/jnanda3/index.html (accessed 01/08/2020.

[67] R. Solera-Urena, A. I. Garcia-Moral, C. Pelaez-Moreno, M. Martinez-Ramon, and F. Diaz-de-Maria, "Real-Time Robust Automatic Speech Recognition Using Compact Support Vector Machines," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 4, pp. 1347-1361, 2012, doi: 10.1109/TASL.2011.2178597.

[68] Z. Zhengyu, G. Jianfeng, F. K. Soong, and H. Meng, "A Comparative Study of Discriminative Methods for Reranking LVCSR N-Best Hypotheses in Domain Adaptation and Generalization," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 14-19 May 2006 2006, vol. 1, pp. I-I, doi: 10.1109/ICASSP.2006.1659977.

[69] E. Dikici, M. Semerci, M. Saraclar, and E. Alpaydin, "Classification and Ranking Approaches to Discriminative Language Modeling for ASR," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 2, pp. 291-300, 2013, doi: 10.1109/TASL.2012.2221461.

[70] S.-O. Caballero-Morales, "Estimation of phoneme-specific HMM topologies for the automatic recognition of dysarthric speech," (in eng), *Comput Math Methods Med,* vol. 2013, pp. 297860-297860, 2013, doi: 10.1155/2013/297860.

[71] A. Garg and D. Roth, "Understanding probabilistic classifiers," in *European Conference on Machine Learning*, 2001: Springer, pp. 179-191.

[72] P. Gautam and S. Soni, "Efficient Speech Recognition with Hidden Markov Models," *International Journal of Advanced Research in Computer Science,* vol. 8, pp. 1988-1995, 2017.

[73] N. Kleynhans, F. de Wet, and E. Barnard, "Unsupervised acoustic model training: Comparing South African English and isiZulu," in *2015 Pattern Recognition Association of South Africa and*

Robotics and Mechatronics International Conference (PRASA-RobMech), 26-27 Nov. 2015 2015, pp. 136-141, doi: 10.1109/RoboMech.2015.7359512.

[74] I. Nurma, A. Hidayat, A. Abdullah, and R. M. Awangga, "Feature Extraction Analysis for Hidden Markov Models in Sundanese Speech Recognition," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 16, p. 2191, 10/01 2018, doi: 10.12928/telkomnika.v16i5.7927.

[75] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 1, pp. 30-42, 2012, doi: 10.1109/TASL.2011.2134090.

[76] D. T. Mane and U. V. Kulkarni, "Visualizing and Understanding Customized Convolutional Neural Network for Recognition of Handwritten Marathi Numerals," *Procedia Computer Science,* vol. 132, pp. 1123-1137, 2018/01/01/ 2018, doi: https://doi.org/10.1016/j.procs.2018.05.027.

[77] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[78] K. Cho, A. Courville, and Y. Bengio, "Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks," *IEEE Transactions on Multimedia,* vol. 17, no. 11, pp. 1875-1886, 2015, doi: 10.1109/TMM.2015.2477044.

[79] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep Learning for Environmentally Robust Speech Recognition: An Overview of Recent Developments," *ACM Trans. Intell. Syst. Technol.,* vol. 9, no. 5, p. Article 49, 2018, doi: 10.1145/3178115.

[80] D. Chen and B. K. Mak, "Multitask Learning of Deep Neural Networks for Low-Resource Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, no. 7, pp. 1172-1183, 2015, doi: 10.1109/TASLP.2015.2422573.

[81] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 20-25 March 2016 2016, pp. 4945-4949, doi: 10.1109/ICASSP.2016.7472618.

[82] A. Biswas, F. De Wet, E. Yilmaz, and T. Niesler, "Multilingual Neural Network Acoustic Modelling for ASR of Under-Resourced English-isiZulu Code-Switched Speech," in *Proceedings of Interspeech 2018*, 2018, pp. 2603-2607.

[83] M. Z. Alom *et al.*, "A State-of-the-Art Survey on Deep Learning Theory and Architectures," *Electronics,* vol. 8, p. 292, 03/05 2019, doi: 10.3390/electronics8030292.

[84] C. M. Bishop, *Pattern recognition and machine learning*. New York: Springer, 2006.

[85] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR,* vol. abs/1412.6980, 2015.

[86] F. de Wet, N. Kleynhans, D. van Compernolle, and R. Sahraeian, "Speech recognition for under-resourced languages: Data sharing in hidden markov model systems," *South African Journal of Science,* vol. 113, no. 1, pp. 25-33, 2017, doi: http://dx.doi.org/10.17159/.

APPENDIX

A. Word Boundary Detection Code

Initialization

```
fs=16e3;
filepath = "File Directory";                %fetch audio file
frameduration = 0.025;                      %set window duration in seconds
frameoverlap = (0.5*frameduration);         % set overlap period to 50% of Frame
duration
file= audioread(filepath);                  %convert audio file to array
windowLength = round(frameduration*fs);      %convert frame duration to sample
format
overlapLength = round(frameoverlap*fs);     %convert overlap duration to sample
format
hopLength = windowLength - overlapLength;
```

Compute Pitch Frequency

```
[f0,loc] = pitch(file,fs, ...
    'WindowLength',windowLength, ...
    'OverlapLength',overlapLength, ...
    'Range',[50 500], ...
    'MedianFilterLength',5, ...
    'Method','NCF');
```

Voice activity detector and zero line crossing per frame

```
buffer = dsp.AsyncBuffer(numel(file));      % create a FIFO buffer equal to the
length of the audio file array length
write(buffer,file);                          %fill buffer with values in file
VAD = voiceActivityDetector;               %function acronym for this particular
code
zcd = dsp.ZeroCrossingDetector;
n = 1;
probabilityVector = zeros(numel(loc),1);    %creat VAD array = length of frame
location
numZeroCross = zeros(numel(loc),1);
while buffer.NumUnreadSamples >= hopLength
    if n==1 % is n = 1?
        x  =  read(buffer,windowLength);          %first  sample  -  no  overlap
information required
           else
        x = read(buffer, ...                %subsequent frames and ovelaps
            windowLength, ...
            overlapLength);
    end
    [pv nv]= VAD(x);
    probabilityVector(n) =pv ;              %record VAD results into an array
    numZeroCross(n)= zcd(x);
    n = n+1;                                 %increase n for the next denoting
the followin frame
end
```

Log energy

```
[s,w,tP,p] = spectrogram(file, ...
                         windowLength, ...       % = 0.025 ms
                         overlapLength,100,fs);  % = 0.0125 ms
Pdb=max(10*log10(p));
```

Spectral Kurtosis

```
[kurtosis,spread,centroid] = spectralKurtosis(file,fs, ...
                             "Window",hamming(round(fs*0.025)), ...
                             "OverlapLength", round(fs*0.0125));
```

Spectral Skewness

```
skewness = spectralSkewness(file,fs, ...
                    'Window',hamming(round(0.025*fs)), ...
                    'OverlapLength',round(0.0125*fs), ...
                    'Range',[62.5,fs/2]);
```