ALGORITHM FOR KEYWORD SPOTTING WITH APPLICATION TO SPEECH RECOGNITION

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Electronics & Communication Engineering

by

Vijayendra A. Desai

119997111012

under supervision of

Dr. Vishvjit K. Thakar

GUJARAT TECHNOLOGICAL UNIVERSITY

AHMEDABAD

February-2017

ALGORITHM FOR KEYWORD SPOTTING WITH APPLICATION TO SPEECH RECOGNITION

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Electronics & Communication Engineering

by

Vijayendra A. Desai

119997111012

under supervision of

Dr. Vishvjit K. Thakar

GUJARAT TECHNOLOGICAL UNIVERSITY

AHMEDABAD

February-2017

DECLARATION

I declare that the thesis entitled "***Algorithm for Keyword Spotting with Application to Speech Recognition***" submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from Sept 2011 to Feb 2016 under the supervision of **Dr. Vishvjit K. Thakar** and this has not formed the basis for the award of any degree, diploma, associateship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of the Research Scholar: ..... Date: 13th February 2017

Name of Research Scholar: **Vijayendra A. Desai**

Place: Ahmedabad

CERTIFICATE

I certify that the work incorporated in the thesis ***"Algorithm for Keyword Spotting with Application to Speech Recognition"*** submitted by **Mr.Vijayendra A. Desai** was carried out by the candidate under my supervision. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/ diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the research scholar.

Signature of Supervisor: ..... Date: 13th February 2017

Name of Supervisor: **Dr. Vishvjit K. Thakar**

Place: Ahmedabad

Originality Report Certificate

It is certified that PhD Thesis titled ***"Algorithm for Keyword Spotting with Application to Speech Recognition"*** by **Vijayendra A**. **Desai** has been examined by us. We undertake the following:

a. Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.
b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results, or words of others have been presented as an Author own work.
c. There is no fabrication of data or results which have been compiled / analyzed.
d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
e. The thesis has been checked using **Turnitin** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time.

Signature of the Research Scholar: …… Date: 13th February 2017
Name of Research Scholar: **Vijayendra A. Desai**
Place: Ahmedabad

Signature of Supervisor: …… Date: 13th February 2017
Name of Supervisor: **Dr. Vishvjit K. Thakar**
Place: Ahmedabad

VIJAYENDRA PHD THESIS

ORIGINALITY REPORT

%17	%6	%6	%16
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Institute of Graduate Studies, UiTM Student Paper	%7
2	Seman, Noraini, Zainab Abu Bakar, and Nordin Abu Bakar. "Measuring the performance of isolated spoken Malay speech recognition using Multi-layer Neural Networks", 2010 International Conference on Science and Social Research (CSSR 2010), 2010. Publication	%3
3	Submitted to Institute of Technology, Nirma University Student Paper	%2
4	www.ukessays.com Internet Source	%2
5	Noraini Seman. "An evaluation of endpoint detection measures for malay speech recognition of an isolated words", 2010 International Symposium on Information Technology, 06/2010 Publication	%1

6	files.gtu.ac.in Internet Source	%1
7	Noraini Seman. "Evaluating endpoint detection algorithms for isolated word from Malay parliamentary speech", 2010 International Conference on Information Retrieval & Knowledge Management (CAMP), 03/2010 Publication	%1

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 1%
EXCLUDE BIBLIOGRAPHY	ON		

PhD THESIS Non-Exclusive License to GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, **Vijayendra A. Desai** (**119997111012)** hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c) GTU is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e) I undertake to submit my thesis, through my University, to any Library and Archives. Any abstract submitted with the thesis will be considered to form part of the thesis.

f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

g) If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h) I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this non-exclusive license.

j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of the Research Scholar: …… Date: 13th February 2017

Name of Research Scholar: **Vijayendra A. Desai**

Place : Ahmedabad

Signature of Supervisor: …… Date: 13th February 2017

Name of Supervisor: **Dr. Vishvjit K. Thakar**

Place: Ahmedabad

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Shri / Smt. / Kum. **Mr Vijayendra A. Desai** (Enrollment No. **119997111012**) entitled **"Algorithm for Keyword Spotting with Application to Speech Recognition"** was conducted on **13 February 2017** at Gujarat Technological University.

(Please tick any one of the following options)

□ We recommend that he/she be awarded the Ph.D. Degree.

□ We recommend that the *viva-voce* be re-conducted after incorporating the following suggestions:

□ The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the Ph.D. Degree.

Name and Signature of Supervisor with Seal

1) External Examiner 1 Name and Signature

2) External Examiner 2 Name and Signature

3) External Examiner 3 Name and Signature

ABSTRACT

The speech recognition system is very useful for the interaction between human and machine. Language is one of the barriers that create a hindrance to human to human interactions. In the scenario of arm conflict or natural disasters we need to communicate with speaker of less prevalent languages. Hence, it is very important and useful to develop a speech recognition system for low resource language like Gujarati. Various applications of local language speech recognition are agriculture, automatic telephone system, voice operated services. The creation of language and acoustic re-sources, for any given spoken language, are typically a costly task. For example, a large amount of time and money is required for the proper creation of annotated speech corpora for Automatic Speech Recognition (ASR) and domain-specific text corpora for Language Modelling (LM). Speech corpora/corpus is database of speech audio files and text transcriptions of these audio files in a format that can be used to create Acoustic Models. For proper working of the system, it is required to identify the spoken words from the given speech inputs, i.e. Keyword spotting plays a crucial role. In this thesis, our work focuses on in-ear microphone compared to conventional microphone system to minimize the effects of background noise. In addition to that, we also implement endpoint detection algorithms and tested algorithms to separate the keywords from the silences and other unwanted noises. For feature extraction, we use Real Cepstral Coefficients (RC) and Mel Frequency Cepstral Coefficients (MFCC). We also configured two and three layers of neural networks and tested for word recognition. For Gujarati speech database generation, various factors are considered such as, speakers of various ages (e.g. Child, young, old), gender (e.g., Male, female), accent (kathiyawadi, sortie, ahmedawadi). In future, our keyword spotting algorithm can be used, to drive a robotic arm; hence the speech database has a vocabulary consisting of ten isolated Gujarati words as follows: ડાબી (Left), જમણી (Right), ઉપર (Up), નીચે (Down), આગળ (Forward), પાછળ (Backward), આજુ (This Side), બાજુ (That side), આમ (Here), તેમ (There).

Acknowledgement

I wish to express my sincere appreciation to those who have contributed to this thesis and supported me in one way or the other during this amazing journey.

First, I am extremely grateful to my supervisor, ***Dr. Vishvjit K. Thakar***, Professor and head of the Department of Electronics and Communication, ADIT college, New Vallabh Vidhyanager, GUJARAT for his guidance and all the useful discussions and brainstorming sessions, especially during the difficult conceptual development stage. His deep insights helped me at various phases of my research. His invaluable suggestions and constructive criticisms from time to time enabled me to complete my work successfully.

The completion of this work would not have been possible without, the Doctorate Progress Committee (DPC) members: ***Dr. Kiran Parmar***, Retired Adjunct Professor, GEC, Gandhinagar and ***Dr. Tanish H. Zaveri,*** Professor, Department of Electronics and Communication, Nirma University, Ahmedabad. I am thankful for their rigorous examinations and precious suggestions during my research.

My gratitude goes out to the assistance and support of ***Dr. Akshai Aggarwal***, Ex. Vice Chancellor, ***Dr. Navin Sheth r***, Vice Chancellor, ***Shri J. C. Lilani***, Registrar, ***Mr. Dhaval Gohil***, Data Entry Operator and other staff members of PhD Section, GTU.

At this stage, I would also like to acknowledge the guidance and support provided by each and every member of ***C. K. Pithawalla College of Engineering and Technology, Surat***. Without that it may not be possible to reach at this stage of my journey in the field of research.

Finally, I would like to thank my mother ***Mrs. Jashumati Vashi*** and my father ***Mr Arvindkumar Ratilal Desai***. They supported me without questioning any of the decisions I made throughout this process. They were always unconditional in extending their trust and belief in me. I would also like to thank my beloved wife ***Vishruti Desai*** and my lovely daughter ***Yashi Desai*** for unconditional love and support in my hard times during this journey. I owe everything to them, without their everlasting love, this thesis would never be completed.

Table of Content

List of Abbreviation

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
CG	Conjugate Gradient
CI	Confidence Interval
DTW	Dynamic Time Warping
EEF	Energy Entropy Feature
$E_{silence}$	Energy of Silence period of speech
HMM	Hidden Markov Model
IIR	Infinite Impulse Response
ITU	Upper Threshold Level
LM	Language Modelling
LM	Levenberg-Marquardt
Logsig	Log Sigmoid Function
LTU	Lower Threshold Level
MFCC	Mel Frequency Cepstral Coefficients
MSE	Mean Square Error
RC	Real Cepstral coefficients
RMS	Root Mean Square (RMS)
SNR	Signal to Noise Ratio
STE	Short Term Energy
Tansig	Tangent Sigmoid Function
ZCR	Zero Crossing Rate

List of Figures

List of Tables

CHAPTER 1

Introduction

Speech is the most natural way to communicate between the humans. The main goal of the speech recognition is to provide the interface between human and machine by recognizing the spoken word or sentence and to respond them [Deller Jr, J. R., Proakis, J. G., & Hansen, J. H. (1993)]. Although it looks like a simple problem, but researcher is working for many decades to get accurate results, because it has multi-dimensional difficulties such as: non-stationary nature of the speech, vocabulary size, speaker dependency [Deller Jr, J. R., Proakis, J. G. & Hansen, J. H. (1993)].

Speech recognition systems are mainly classified based on the:

1. Types of input speech
2. Types of feature extraction and classification methods

In speech processing system input speech, may be continuous, spontaneous, dictated sentences or isolated words. In the current study, isolated speech input is considered.

1.1 Speech recognition for Indian language

A speech recognition system for the Indian language, specifically for the Gujarati language was the primary goal of the research. At events, such as natural disasters and social gathering, it is required to communicate with the speakers of less prevalent languages. As per the survey done for census 2011, India has 122 major languages and 2371 dilates [Rohini B Shinde and V P Pawar (2012)]. More interestingly accents are not same in language speaking society [Harisha, S. B., Amarappa, S., & Sathyanarayana, D. S (2015)]. This is a major hurdle to develop Automatic Speech Recognition (ASR) system for Indian languages. Another issue is, most of the Indian languages are low resource languages. As per the Krauwer [Krauwer, S. (2003)], language is considered to be a low resource if it falls into the following criteria: lake of

unique writing system or orthography, limited presence on the web, lack of linguistic expertise, lack of electronic resources such as: monolingual corpora, bilingual electronic dictionaries, transcribed speech data, vocabulary list [Krauwer, S. (2003)]. Various applications of the local language speech recognition are agriculture, automatic telephone system, voice operated services. The largest amount of time and money is required for the proper creation of annotated speech corpora for automatic speech recognition (ASR).

1.2 In-ear microphone concept

The source of speech input also plays crucial role in the accuracy of the recognition system. For lab experiments we can use pre-recorded speech samples. These samples are recorded in studio type of environment, where minimum chances of background noises and other interfering factors. But for real time application, speech inputs are taken in a traffic area, outdoor environment where maximum chances of external noises. Even in the case of indoor environment, there are possibility of interference from other speakers or people.

It's important to design a speech recognition system considering all these factors. Until now, speech signal collected through a microphone placed in front of the mouth has been the primary source of the speech recognition applications. The problem associated with this type of speech collection is that the ambient noise is also picked up via microphone at the same time. As a solution of it, there is need to find an alternative way to record a speech in real time situation. The external auditory canal, when isolated properly with an ear-insert microphone can provide intelligible speech even in sever noise condition.

In my work, I have used in-ear microphone, for recording speech inputs. It provides isolation from surrounding noises, as speech in this case passes through the internal cavity between ear and mouth. Results obtained with the in-ear microphone, clearly suggest that its superior compare to conventional microphone system.

1.3 End point detection

For keyword spotting, speech is required to separate from the silence portion. The process of separating speech segment of utterance from the non-speech segment is called endpoint detection. Different endpoint detection systems are tested in current work. Accuracy of the

end point detection is important for two reasons [Ying, Ying, G. S., C. D. Mitchell, and L. H. Jamieson (1993)]:

1. Accurate detection of end point will reduce word error rate.
2. Overall processing of speech recognition becomes faster, as the algorithm will not waste time in non-speech segments.

1.4 Feature extraction and Artificial Neural Network

Features are parametric representation of the speech, and different technique is applied to extract them and to use them efficiently for a speech recognition system. Common features include the Liner Predictive Coding Coefficients (LPCCs), Real Cepstral coefficient (RC) and Mel-frequency Cepstral coefficients (MFCC). The LPCCs and RCs were the most popular choices for the speech recognizer up until the 1980s [Davis, S. B., and Mermelstein, P. (1980)]. Study by Davis and Murmelstein [Davis, Murmelstein] shows superiority in performance of the MFCC versus well-known LPCCs and RCs. The power of the MFCCs comes from the fact that their extraction approximates the human perception. For feature extraction, Real Cepstral coefficient (RC) and Mel-frequency Cepstral coefficients (MFCC) are tested and results are compared.

For classification, different combinations of two and three layers are configured and tested. For the hidden neurons, hyperbolic tangent sigmoid function (tensing) is used, because of its non-linear distribution characteristics. Log sigmoid function (logsig) is used for the output neurons, in order to restrict the network output to the interval [0,1]. Two types of learning algorithms are tested:

1. Conjugate Gradient (CG),
2. Levenberg-Marquardt (LM).

From the results, it has been proved that CG algorithm is computationally much faster and led to better classification results within the minimum memory requirement.

In 1980s, speech recognition system becomes smarter. They are able to predict the word also. The major reason for this upgradation is the use of new statistical method HMM, rather than using simple templates for words and looking for sound patterns. HMM considered the probability of unknown sounds being words. By the group of Carnegie Mellon and IBM

introduced discrete density HMMs, which laid down the foundation of modern HMM based speech recognition system [Lippmann RP (1990)]. Later Bell laboratories introduces continuous density HMM [19-21]. The main reason for this vast improvement in the speech recognition technology is application of fundamental pattern recognition methods based on the LPC developed by Itakura [F. Itakura (1975)], Rabinar Levinson [Rebinar (1989)] and others.

In the early 80's Artificial Neural Network (ANN) technology introduced. The brains impressive superiority at a wide range of cognitive skills, motivated the researcher to explore possibilities of ANN models in the field of speech recognition [Hinton, G. E. (1989).], With the hope that human neural network like models may ultimately lead to human like performance. Early attempts to use neural networks for speech recognition concentrated to few phonemes, few words and few isolated digits, with good success using pattern mapping by multilayer perceptron (MLP).

1.5 Objective and Scope of Work

1 The Gujarati speech database is prepared with peoples of different age, gender and region.

2 In-ear microphone concept is used to minimize the effect of surrounding noise in outdoor recording.

3 Different end point detection methods are tested to separate the spoken words from the silence part.

4 Real Cepstral (RC) and Mel-Frequency Cepstral coefficients (MFCCs) use of the feature extraction methods.

5 Different combinations of two layer and three layer neural networks are configured and tested.

1.6 Thesis Structure

The thesis consists of eight chapters. Chapter 2 contains a literature review. chapter 3 contains in-ear microphone detail. Chapter 4 covers several speech endpoint detection methods. Chapter 5 explains different methods of feature extraction implemented. Chapter

6 contains the detailed description of the recognition methods using two layers and three-layer neural. Recognition results are discussed in chapter 7. Conclusion and future scope is discussed in chapter 8.

CHAPTER 2

Literature survey

The speech recognition system is used to provide interface between human and machine. The machine should be able to understand the speech command given by human for performing different task suggested by human. Different algorithms are suggested and implemented for it. This chapter includes state of the art for speech recognition system.

2.1 Fundamentals of Speech Recognition System

Despite of years of research speech recognition system is still challenging field. There are multiple factors which affects the performance of the speech recognition accuracy. Such as, background noise effect, speaker variability, same word spoken differently by different region of people with in same county like India, types of words i.e isolated, continuous, dictation type. So, various researcher works take into consideration during literature survey. Speech is the most natural way for communication between different people. The aim of speech recognition system is to make interaction between human and machine possible [Deller Jr, J. R., Proakis, J. G., & Hansen, J. H. (1993)]. It seems to be a straightforward problem, but from a researcher it has been revealed that it's difficult to achieve accurate results. The speech recognition system faces multidimensional problems such as non-stationary nature of speech, large vocabulary size, confusable words, speaker dependency, large processing time [Deller Jr, J. R., Proakis, J. G., & Hansen, J. H. (1993)]. Speech recognition systems are mainly classified based on the:

1 Type of speech input,

2 Type of feature extraction and classifications method.

Based on the type of speech input, recognition systems are classified as

1 Isolated word system,

2 Continuous word system

3 Spontaneous word system

4 Speaker dependent system

5 Speaker independent system

Isolated word speech recognition system is considered for the current study. Isolated word recognition system is having a prescribed recording intervals, consists of an isolated word preceded and followed by silence or other background noises [Lamel, L. F., Rabiner, L. R., Rosenberg, A. E., & Wilpon, J. G. (1981)]. The process of separating the speech segments of an utterance from the non-speech segment obtained during the recording process is called end point detection. Accurate detection of end points is important for two reasons [Ying, G. S., Mitchell, C. D., & Jamieson, L. H. (1993)]:

Overall speech recognition accuracy depends on accurate end point detection. Computation for processing the speech is minimum when the endpoints are accurately located.

Based on the feature extraction and classification approaches used for speech recognition, ASR systems are classified as:

1. Acoustic Phonetic system
2. Pattern Recognition system
3. Artificial Intelligence system

Acoustic Phonetic Approach: The earlier approaches of speech recognition were based on finding the speech phonetic characteristics using spectral analysis and to provide the appropriate labels to these sounds [Rabiner, L. R. (1989)]. This approach is highly speaker dependent, large data set and language models are required for matching. It is not widely used in the commercial applications.

Pattern Recognition Approach: It mainly consist of two steps: pattern training and pattern comparison [(Itakura 1975; Rabiner 1989; Rabiner and Juang 1993)]. They use well formulated mathematical framework for speech pattern representation and reliable pattern comparison. It mainly consists of two approaches.

1. Template based approach
2. Stochastic approach

Template based Approach: The underlying idea is simple in this approach. A collection of prototypical speech patterns is stored as reference pattern, called as a dictionary or database of candidates' words. Recognition is then carried out by matching an unknown utterance with each of this reference templates and selecting the category best matching pattern. Dynamic Time Wrapping (DTW) is working on this principle [H.Sakoe and S.Chiba (1978)]. Usually templates for the entire words are constructed. It has advantages that, the error due to segmentation and classification of smaller acoustically variable such as phonemes can be avoided. On the other hand, template creation becomes expensive and practically impossible as vocabulary size increases.

Stochastic Approach: It uses probabilistic models to deal with the uncertain or incomplete information. In a speech recognition system uncertainty arises from many sources such as: confusable sounds, speaker variability. Most popular stochastic probability models are Hidden Markov Model(HMM) [L.R.Rabiner,(1989)]. Compare to template based approach, HMM is more general and has better mathematical foundation [L.R.Rabiner,(1989)]. Compared to knowledge based approach, HMM can easily enable integration of knowledge sources in to compiled architecture. However, HMM does not provide much insight of the recognition process, so it is often difficult to analyze errors and to improve its performance.

Artificial Intelligence Approach: It is a hybrid of the acoustic phonetic approach and pattern recognition approach. It is also called as a knowledge based approach, because it exploits information regarding linguistic, phonetic and spectrogram. Earlier this approach had limited success, largely due to the difficulty in qualifying expert knowledge of phonetics, phoneticians, lexical access, syntax, semantic and pragmatics [L.R.Rabiner,(1978)]. During 1980s, Artificial neural networks (ANN) was introduced [J.Ferguson,Ed.,(1980)],"Hidden Markov models for speech," IDA, Princeton, NJ]. The brain impressive superiority at a wide range of cognitive skills, has motivated the researcher to explore the possibilities of ANN in the field of speech recognition [Hinton, G. E. (1989)] with hope that human neural network like models may ultimately lead to human like performance. With advancement in the high-speed computer processor, we can achieve good parallel processing, required for ANN to do bottom up and top down processes between feature, phoneme and word level to recognize

presented word. By using different configuration of input and output combinations along with hidden layers we are able to achieve good accuracy of word recognition.

2.2 Speech Recognition for Indian Languages

As per the Global Monitoring Report (GMR) released by the UNESCO, India has largest population of illiterate adults in the world with 287 million [Ojanen, E., Ronimus, M., Ahonen, T., Chansa-Kabali, T., February, P., Jere-Folotiya, J., & Puhakka, S. (2015)]. So, it is difficult for them to work directly with machinery, which required text inputs or understand only English language. In this kind of situation for human machine interface speech recognition system is very useful.

It is a challenging problem, as Linguistic diversity is very rich and wide in India. According to census 2011, India has 122 major languages and 2371 dialects. Out of 122 languages 22 are constitutionally recognized languages [Hemakumar, G., & Punitha, P. (2013)]. In Indian languages, some of them have many scripts and other may have only one script. More interestingly accent is not uniform within the same language speaking society [Hemakumar, G., & Punitha, P. (2013)]. This is a major hurdle to develop the Automatic Speech Recognition (ASR) system for Indian Languages. Another major issue is, most of the Indian languages are low resource languages.

The term Low resource language is first introduced by Krauwer [Krauwer, S (2003)] refers to language with some of the following aspects: lake of unique writing system or orthography, limited presence on the web, lack of linguistic expertise, lack of electronic resources for speech and language processing, such as monolingual corpora, bilingual electronic dictionaries, transcribed speech data, pronunciation dictionaries, vocabulary lists [Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014)]. It is important to note that it is not similar to a minority language, which is spoken by minority of the population of a territory. In order to objectively define the status of a language, the concept of BLARK (Basic LAnguage Resource Kit) was defined in a joint initiative between ELSNET (European Network of Excellence in Language and Speech) and ELRA (European Language Resources Association) [Krauwer, S. (2003)].

Annotated Speech Corpora of approximately 50 hours were developed for Hindi, Marathi, Punjabi, Bengali, Assamese, Manipuri, Tamil, Malayalam, Telugu and Kannada [Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014)]. However, the corpora are not provided to public for research activities and commercial purpose. As per the survey of the work done in the Indian languages, it's found out that limited work is done in the field of the Gujarati speech recognition system.

In the paper of Himanshu N. Patel [Patel, H. N. (2015)]''Automatic text conversion of continuous speech for Indian languages", the training data for the Gujarati language is aligned using an existing speech recognition engine for English language. This aligned data is used to obtain the initial acoustic models for the phones of the Gujarati language. So, it's not for keyword spotting algorithm, but it's mainly related in direction of text to speech conversion.

In the paper of Jigarkumar Patel''Development and Implementation of Algorithm for Speaker recognition for Gujarati Language" [Patel, J., & Nandurbarkar, A. (2015)], the work is done for speaker recognition using Gaussian Mixture Model (GMM). They have suggested modification in Mel Frequency Cepstral Coefficient (MFCC). Work is concentrated in the field of the speaker recognition system.

In the paper of Jinal H. Tailor 'Speech Recognition System Architecture for Gujarati Language" [Tailor, J. H., & Shah, D. B. (2016)] suggested speech to text system for Gujarati language using Hidden Markov Model (HMM).

CHAPTER 3

In-Ear Microphone and Speech Database

3.1 Speech Data Set Generation

Speech collection is an important part of the speech recognition system. In almost all types of Speech recognition system, speech is collected by keeping the microphone outside the mouth. The perfect studio environment is required for good quality and noise free speech recording. Majority application of speech recognition system working in real time situation where it is required to record speech signal in heavy road traffic, office environment and similar kind of situation where the maximum probability to pick up surrounding noises, if high sensitive microphone is used then. It may also pickup user generated artifacts such as heavy breath coughing. In case of low sensitive microphone, it may miss some of the speech samples of the user. As a solution of this problem, in my work, speech is collected from the ear, using in-ear microphone. Due to human body structure, ear and mouth are connected through the internal cavity structure. So, a portion of the speech is produced by the speech production system of the human body also available in the human ear system. From the results its observed that, by keeping slightly higher sensitive microphone in the ear, we are able to capture good quality speech signal with minimum amount of external noise. Results also show that, internal structure of the human body works as low pass filter so, speech signals are damped for frequency above 8.7 kHz. The in-ear microphone is mounted with cotton, which serves two purposes: it allows microphone to stand still in the ear and it closes the external auditory canal to provide shielding from ambient noises. So, improves overall speech quality.

For Gujarati speech database generation, using in-ear microphone, various factors are considered such as, speakers of various ages (e.g. Child, young, old), gender (e.g., male, female), accent (kathiyawadi, surti, ahmedawadi). In future, our keyword spotting algorithm can be used, to drive a robotic arm; hence the speech database has a vocabulary consisting

of ten isolated Gujarati words as follows: ડાબી (Left), જમણી (Right), ઉપર (Up), નીચે (Down), આગળ (Forward), પાછળ (Backward), આજુ (This side), બાજુ (That side), આમ (Here), તેમ (There).

TABLE 3.1 Total Speech Database

Speaker number	NUMBER OF REPEATION WORDS									
	ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આમ	તેમ	આજુ	બાજુ
1.	6	8	8	9	6	9	9	8	8	8
2.	9	6	8	9	6	9	8	9	8	6
3.	9	9	8	6	8	8	9	8	9	9
4.	6	8	6	8	9	9	8	9	8	6
5.	9	9	6	8	9	8	9	8	9	9
6.	9	9	8	9	6	8	6	8	6	8
7.	8	8	9	8	9	9	8	9	8	9
8.	9	8	6	8	8	8	8	8	8	8
9.	8	9	9	8	6	9	6	9	9	9
10.	9	8	9	8	9	8	9	8	9	8
11.	9	8	9	9	8	9	8	9	8	9
12.	9	9	8	9	8	8	9	9	8	9
13.	6	8	6	8	6	8	6	8	9	6
14.	9	8	6	8	8	6	9	6	9	6
15.	8	9	8	9	8	8	6	9	9	8
16.	8	6	8	9	8	9	8	6	8	9
17.	8	9	8	9	8	9	8	9	8	9
18.	8	9	8	9	8	9	9	8	9	8
19.	8	8	9	8	9	8	9	8	9	8
20.	9	8	9	8	9	8	9	8	9	6
21.	9	6	9	6	9	9	9	7	7	9
22.	9	7	9	9	9	9	8	8	7	8
23.	9	8	9	7	7	8	8	7	9	9
24.	9	9	8	8	8	7	9	6	9	6
25.	6	8	9	8	8	9	8	9	8	9
26.	8	9	8	9	8	9	8	9	8	9

27.	8	6	8	6	8	8	9	8	8	9
28.	8	9	8	8	9	9	8	8	8	9
29.	9	8	9	8	8	9	8	8	9	8
30.	8	8	8	8	6	7	7	8	6	8
31.	8	8	9	8	9	8	8	9	8	9
32.	8	9	8	8	9	8	9	8	9	8
33.	8	9	8	9	8	8	9	8	8	8
34.	8	8	9	8	9	6	8	7	7	8
35.	8	8	9	8	7	9	6	8	9	8
36.	8	9	9	8	8	8	8	8	8	8
37.	8	8	8	8	8	8	8	8	8	8
38.	8	7	8	6	9	8	7	8	8	8
39.	7	8	8	7	8	7	8	8	9	8
40.	9	8	7	8	9	8	7	9	6	7
Total word	327	324	324	322	320	329	321	323	327	322
Average of words	8.175	8.1	8.1	8.05	8	8.225	8.025	8.075	8.175	8.05
Overall data size	3239									

Using MATLAB programing each recorded speech is converted to text files, and stored in the individual folders of speaker and subfolders of the ten words. Each speaker speaks same words for the ten times, to improve the overall database of the speech signal. Total 40 speakers speak 70 different words with same words spoken 70 times, so the total speech dataset generated is: 40*70*70= 4000. All recordings belong to same speaker is stored into one folder. Next each word file is split into individual trials in separate folders for easy access and manipulation. Using end point detection, each word boundary is detected and cropped words are stored in the individual folders and sub folders.

For comparison with speech generation using in-ear microphone and conventional recording system, two words are recorded by keeping the microphone outside mouth for all users with ten times each. So, it adds 40*8*70=800 more words to speech dataset collections.

During endpoint detection, speech signals are tested with different algorithms. Based on this, some of the words are discarded if they not satisfied suitable conditions. So, overall speech

dataset values may not be fixed. Table 1 shows the total speech dataset used for current work.

Speech is produced by the movement of the human articulation system consists of the vocal tract, tongue, lips, teeth and air force from the lungs. Due to rapid movements of articulation system, speech is time varying. Direct analysis of the speech would be difficult, so speech, is segmented into the smaller units or sounds that have certain articulatory property, called as phones.

Phonetic classifications are the process of grouping phonemes based on their properties related to the waveforms, frequency characteristics, manner of articulations, place of articulations, types of excitation and stationary characteristics. \cite (ref Deller, Proakis, Hansen 1993). The speech dataset used in this work is very rich in terms of the phonetic distribution as shown in Table 2. It consists of different types of vowels, fricatives, stops (plosive), nasals, liquids and diphthongs.

3.2 Spectral Characteristics of the Speech Data

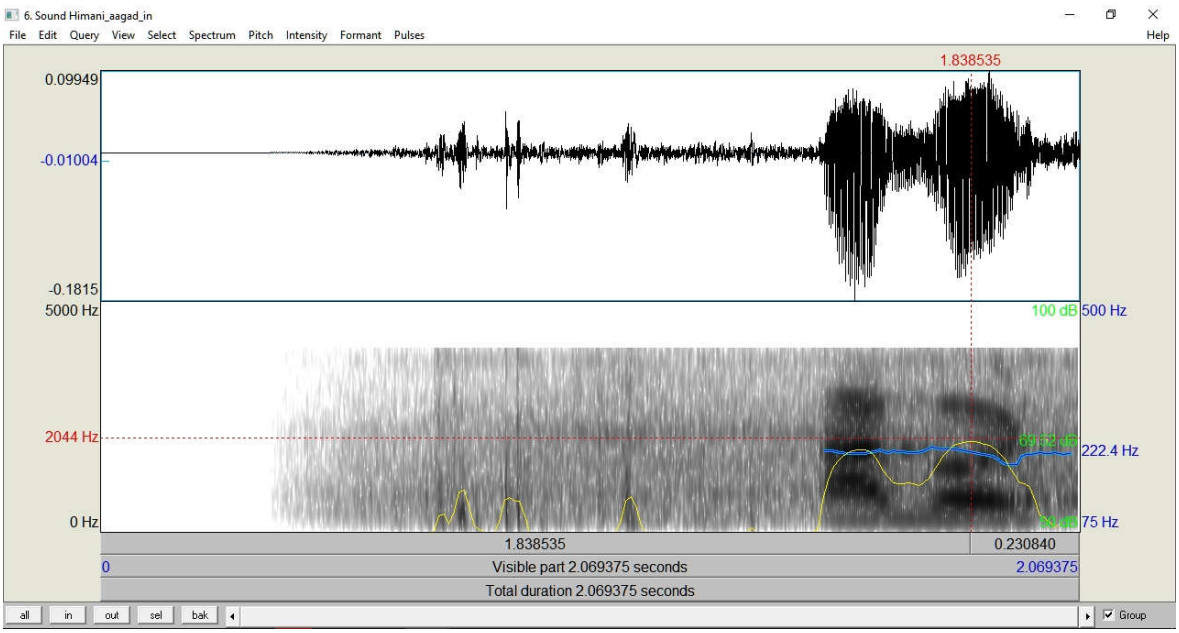


FIGURE 3.1: Waveform and spectrogram for the word "agad" by keeping in-ear microphone.

Short Time Fourier Transform (STFT) is used, to extract spectral characteristics of the speech dataset. It will be represented by the spectrogram, as speech is combinations of frequency dependent parameters. The recorded utterance is sampled at 8 kHz rate so;

frequency axis of the spectrogram will go up to 4 kHz. The time axis will go up to the length of the recording signal.

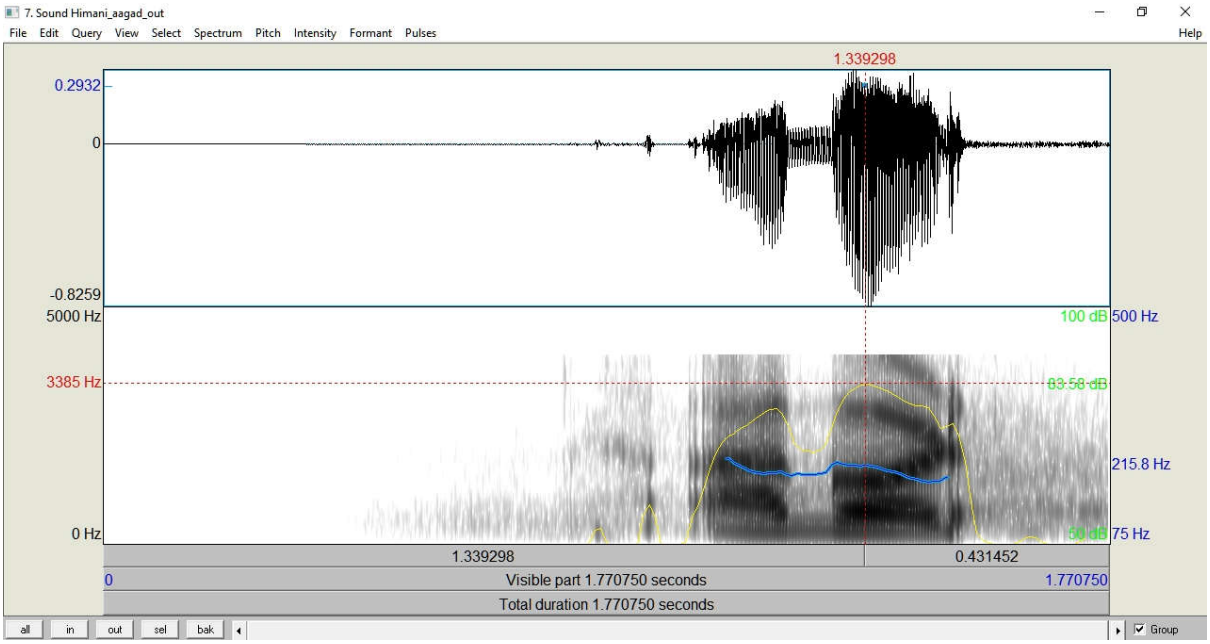


FIGURE 3.2: Waveform and spectrogram for the word "agad" by keeping the microphone outside the mouth.

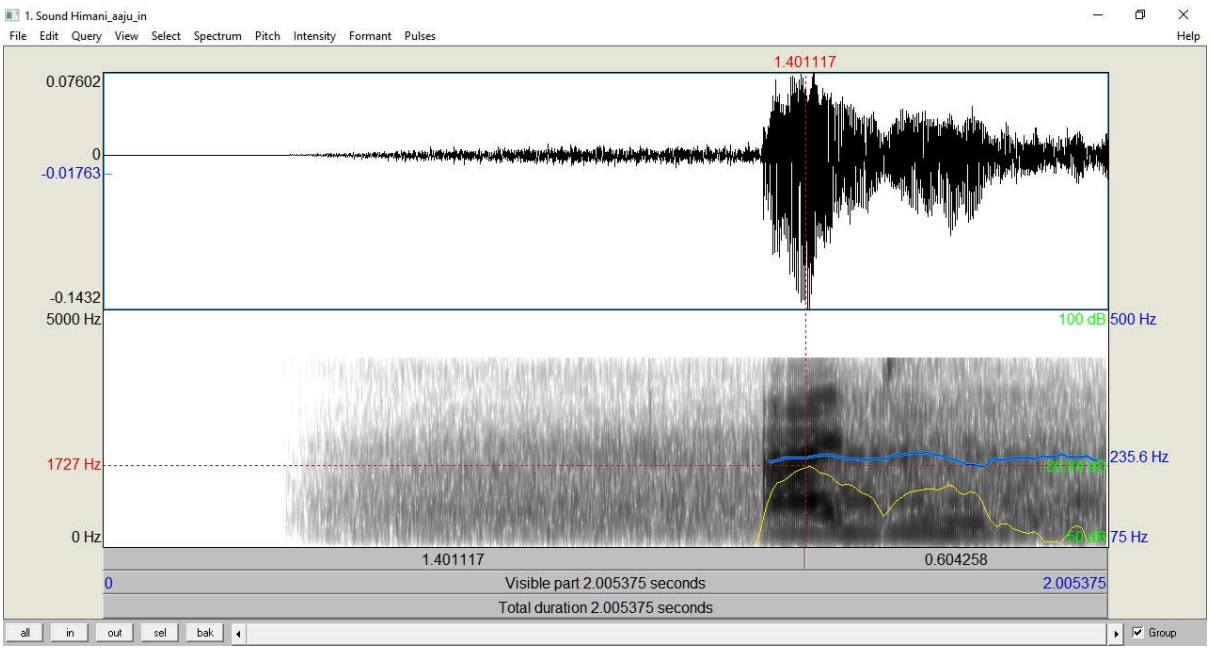


FIGURE 3.3: Waveform and spectrogram for the word "aaju" by keeping in-ear microphone.

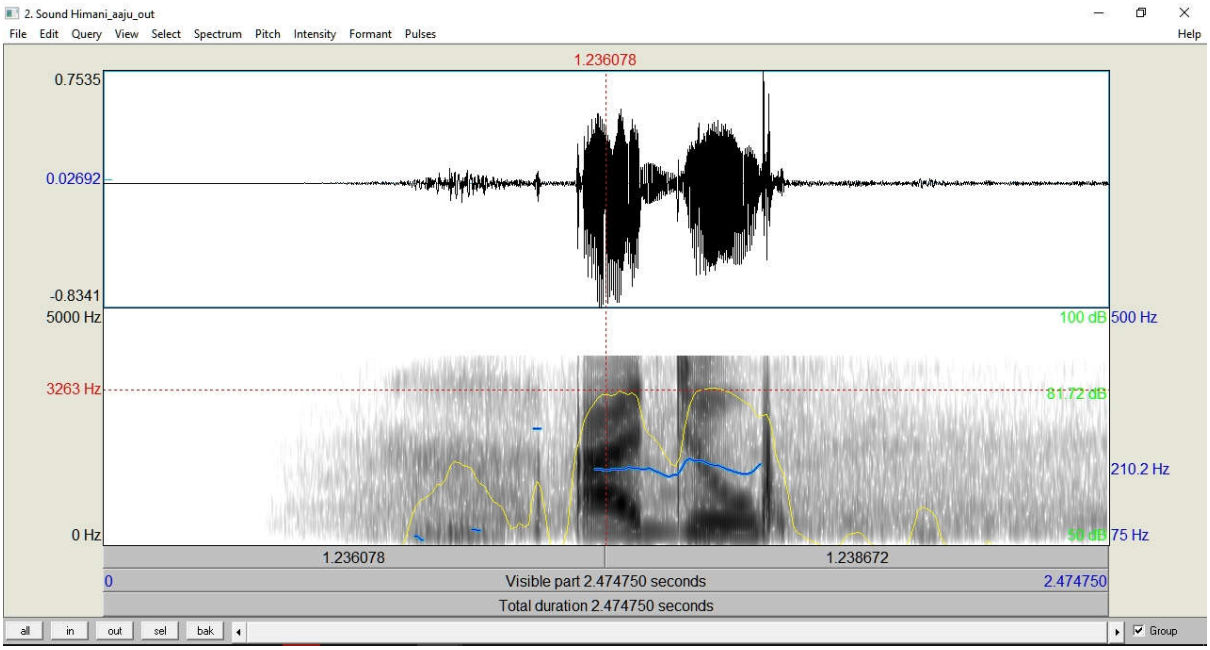


FIGURE 3.4: Waveform and spectrogram for the word "aaju" by keeping the microphone outside the mouth.

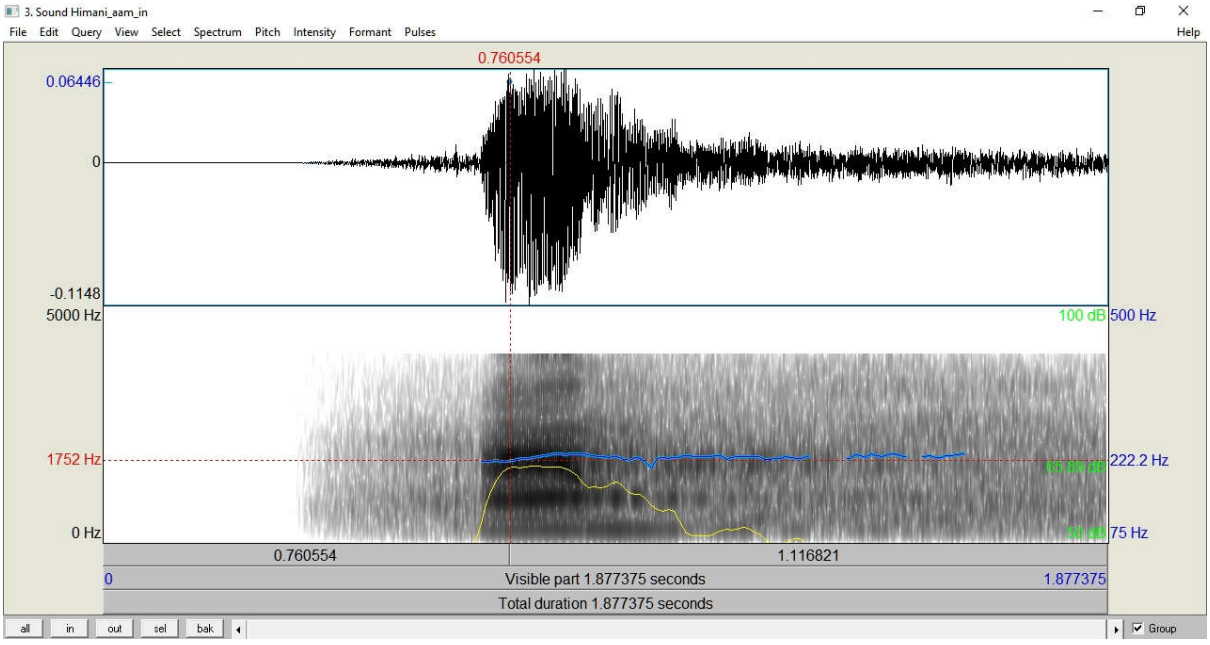


FIGURE 3.5: Waveform and spectrogram for the word "aam" by keeping in-ear microphone.

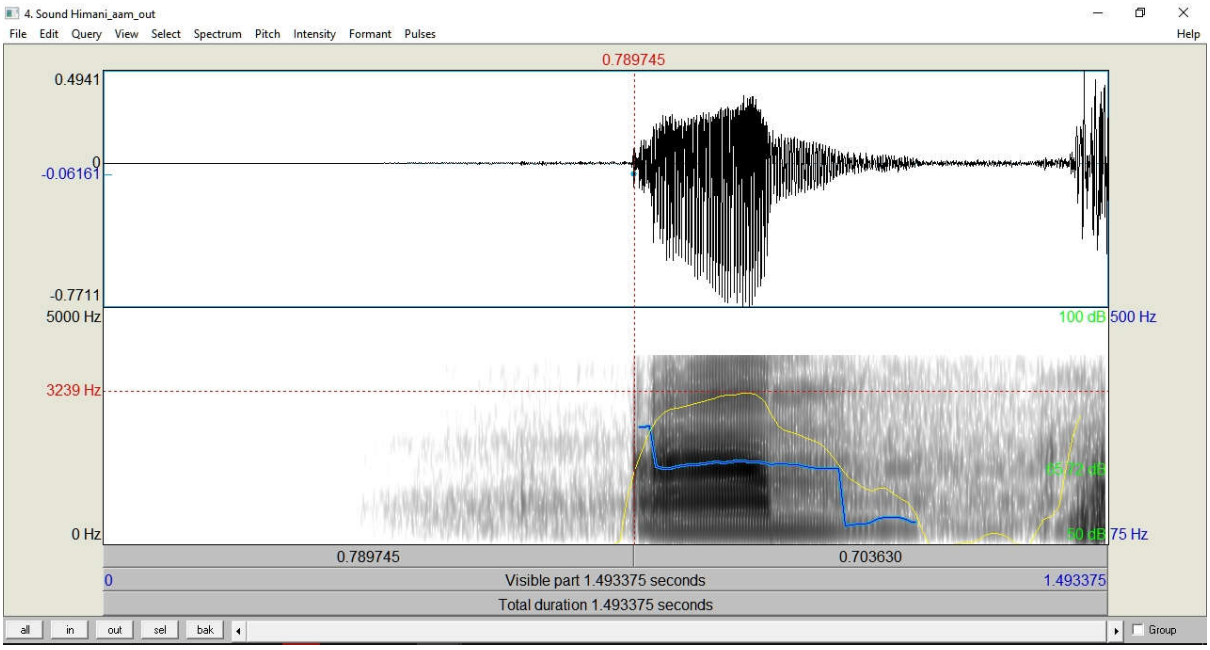


FIGURE 3.6: Waveform and spectrogram for the word "aam" by keeping the microphone outside the mouth.

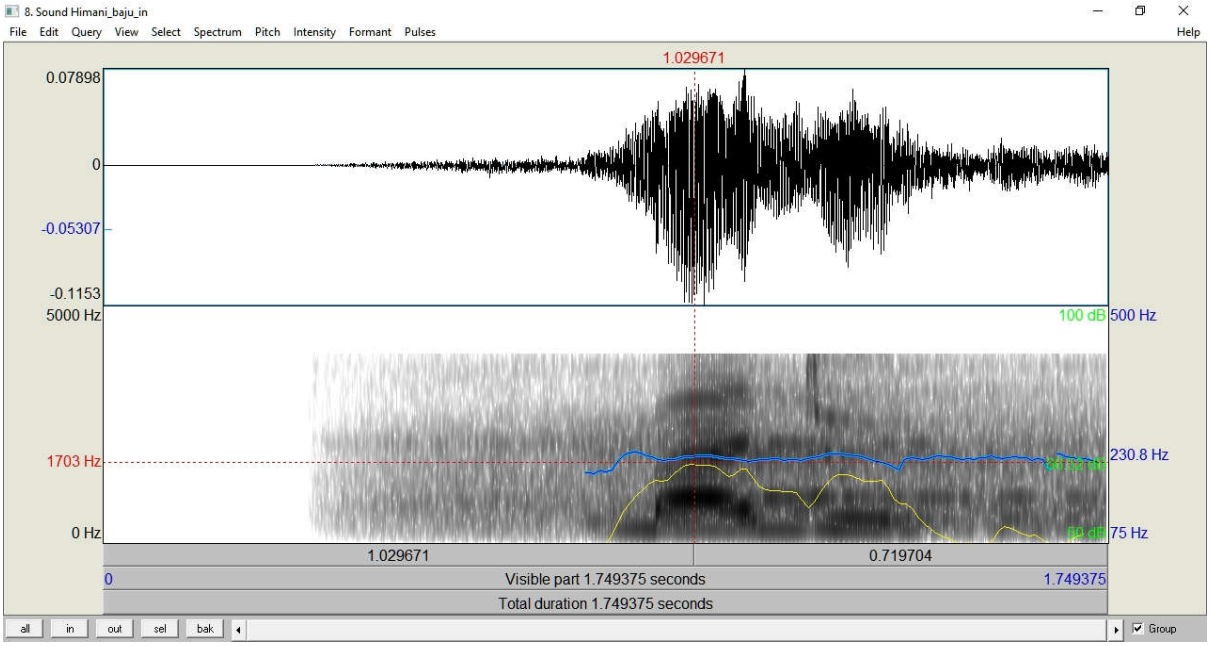


FIGURE 3.7: Waveform and spectrogram for the word "baju" by keeping in-ear microphone.

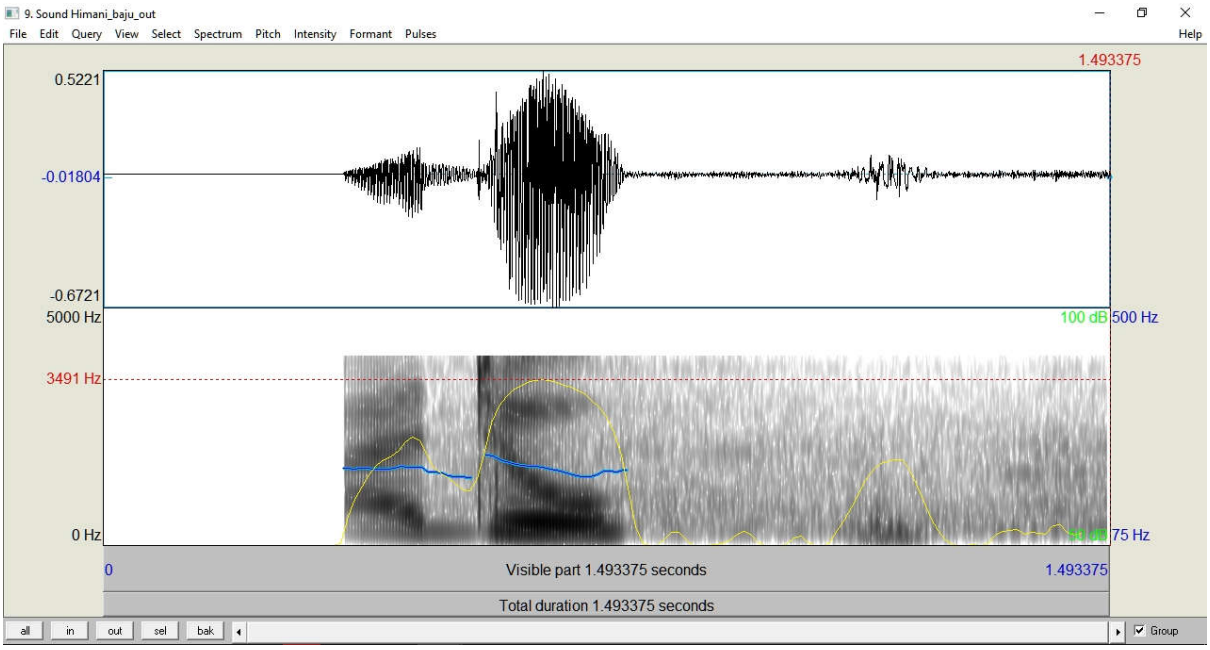


FIGURE 3.8: Waveform and spectrogram for the word "baju" by keeping the microphone outside the mouth.

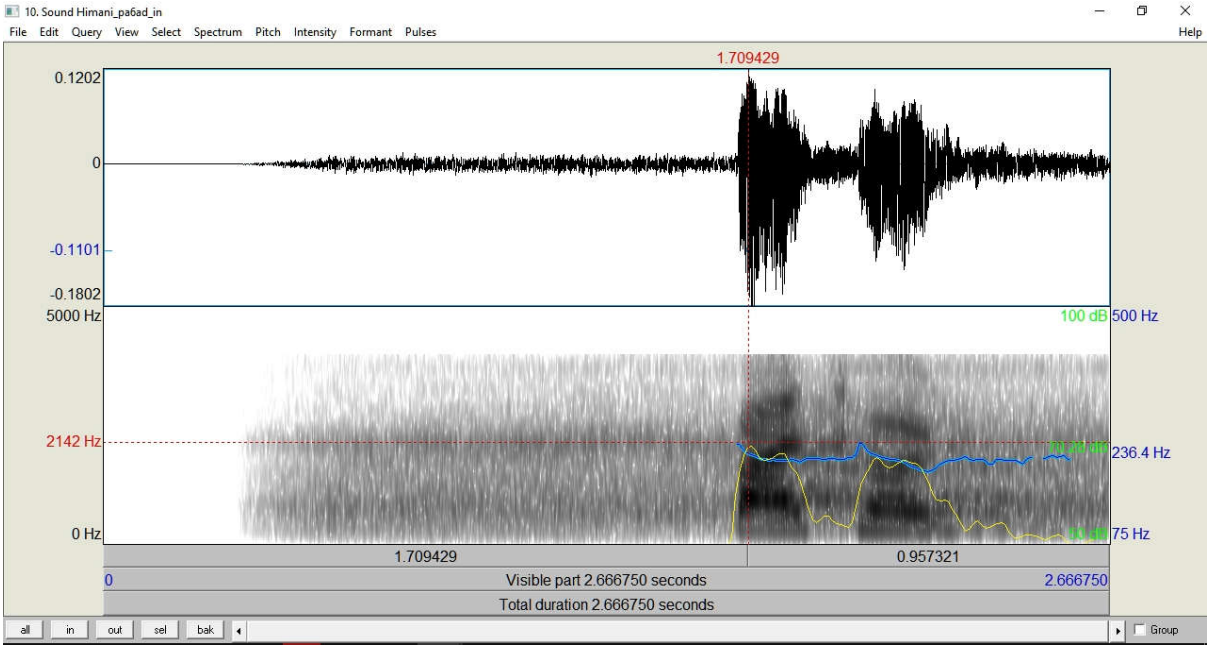


FIGURE 3.9: Waveform and spectrogram for the word "pachhad" by keeping in-ear microphone.

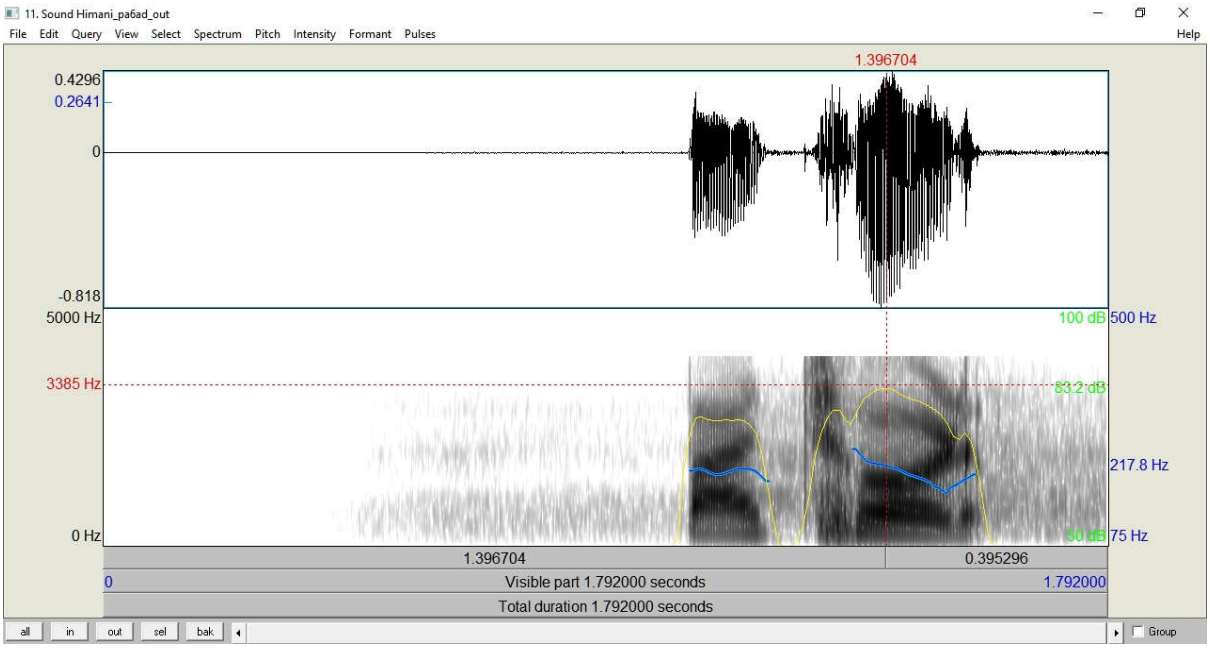


FIGURE 3.10: Waveform and spectrogram for the word "pachhad" by keeping the microphone outside the mouth.

Time domain waveform and spectrogram for different speech signals recorded using in-ear microphone and conventional microphone recording are shown in fig 3.1.

TABLE 3.2: Word List Used and Its Phoneme Distribution.

Word		apabet	Phoneme	class	Manner of articulation	Example word
ડાબી	ડ	da	ə	Consonants	stop	Down
	આ	aa	ā or¨	vowels	mid	After
	બ	ba	bə	Consonants	stop	Bus
	ઈ	i	ī	Vowels	front	In
જમણી	જ	ja	də	Consonants	affricates	Judge
	મ	mas	mə	Consonants	nasals	Mug
	ણ	nba	ə	Consonants	nasals	Number
	ઈ	i	ī	Vowels	front	In
ઉપર	ઉ	u	uw	Vowels	back	Use
	પ	pa	pə or¨	Consonants	stop	Pub
	ર	r	r	semivowels	glides	Run
નીચે	ન	na	nə	Consonants	nasals	Ninja
	ઈ	i	ī	Vowels	front	In
	ચ	cha	tə	Consonants	affricates	Chair
	એ	e	e,	Vowels	front	Egg
આગળ	આ	aa	ā or¨	vowels	mid	After
	ગ	ga	ə	Consonants	stop	Gum
	ડ	da	ə	Consonants	stop	Done
પાછળ	પ	pa	pə	Consonants	stop	Park
	છ	chha	tə	Consonants	affricates	
	ડ	da	ə	Consonants	stop	Done
આજુ	આ	aa	ā or¨	vowels	mid	After
	જ	ja	də	Consonants	affricates	Judge
	ઉ	u	uw	Vowels	back	Use
બાજુ	બ	ba	bə	Consonants	stop	Basket
	આ	aa	ā or¨	vowels	mid	After
	જ	ja	də	Consonants	affricates	Judge
	ઉ	u	uw	Vowels	back	Use
આમ	આ	aa	ā or¨	vowels	mid	After
	મ	mas	mə	Consqgants	nasals	Must
તેમ	ત	ta	tə	Consonants	stop	
	એ	e	e,	Vowels	front	Egg
	મ	mas	mə	Consonants	nasals	Must

3.3 Summary

This chapter presented the main idea behind the speech dataset collection using in-ear microphone. It also explained the phonetic and spectral properties of the vocabulary word selected for the dataset generation.

Few important points are observed from the waveforms.

1 There is a constant low frequency hum is present in all the spectrogram.

2 This hum occupied the frequency range from 0 to 100 Hz in the case of in-ear microphone recording.

3 In case of speech recorded by keeping the microphone outside mouth this, hum value will go up to 1.25 kHz.

4 The voice portion of the speech signal where maximum energy is concentrated, have frequency value mainly up to the 2.3 KHz.

5 The frequency content of the word collected via an outside microphone goes up to 3.5 KHz.

6 So, we can say that in-ear microphone behaves like a low pass filter which suppress the high frequency noise signal.

7 The in-ear microphone placed into the ear canal picks up mainly the bone conducted speech and sound conducted through the muscles and tissues covering the skull.

CHAPTER 4

End Point Detection

End point detection is the most important part of any speech recognition system. This chapter first present basic idea behind end point detection system. Second, it presents the problems encountered in the end point detection. Third, different parameters and features commonly employed for various end point detection system such as short-time energy, Teager's energy, zero crossing rate. Fourth, it discusses end point detection algorithm implemented for current study, which uses infinite impulse response (IIR) bandpass filter at the processing stage prior to detection and energy entropy combination for final word boundary detection.

4.1 Endpoint Detection Basics

End point detection is the process of finding edge points of the uttered word or speech segment in the presence of the background noises. It used to find start and end of the spoken word, i.e it's also known as word boundary detection.

Accurate detection of the speech end point is very important for recognition application, for two reasons. [Lamel, Rabiner, Rosebberg Wilpon, 1981]

- Accuracy of the word boundary detection effects overall accuracy of speech recognizer.
- Accurate detection of the end point detection, successfully removes the unwanted portion of the recorded speech, i.e. background noise and silence portion. so, it reduces overall computations significantly.

The first point can be understood very easily, if we take the example of speech recognition system, which uses Dynamic Time Wrapping (DTW) system. With this method in coming end point detected speech signals are compared with the stored templates and based on the matching criteria corresponding word can identify. In this case word boundary detection and

alignment become a crucial issue. The study of Junqua et al. [Junqua, 1991] suggested that more than half of the recognition errors are due to the word boundary detection errors.

The second point is self-explanatory. If we remove unwanted silence and background noise portion, then speech recognizer will not waste time for feature extraction and classifications where speech is not present.

Speech end point detection looks trivial task to achieve, but it is not the case, unless signal to noise ratio is very good for recording speech. Mostly, for real time recording this is difficult to achieve as background noises highly affect the speech signals. So, low signal to ratio and different types of noises makes end-point detection challenging field in real time application. So, End point detection has been active research topic since 1970.

Up till 1990s, many limited works have been done in this field. Two studies were mostly referred by the researcher, suggested by the Rabiner [Rabiner, Sambur, 1975] and Lamel [Lamel, Rabiner, RosenbergmWilpon, 1981], these are mainly based on the energy measures. There is noticeable amount of increasing in the research work after 1990, and many different techniques have been investigated such as spectral approaches, variable frame rate methods and lately entropy based methods. However, this approach is mostly data specific, so not accepted or widely used globally.

Three types of the endpoint detection methods are commonly available: implicit, explicit and hybrid. The main difference between first two methods is, separate endpoint detection stage used in the explicit method. It's found out that explicit methods always give accurate result compared to the other two methods. So, in current work explicit method is applied. The general block diagram of explicit method is shown in fig 4.1.

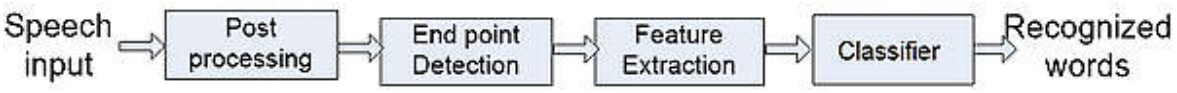


FIGURE 4.1: Block Diagram for The Explicit Speech Recognition System

End point detection algorithms suggested so far are based on the short-term energy, spectral energy like Teager's energy, zero crossing rate (ZCR), variable frame rate (VFR) methods using cepstral or time derivatives features, entropy or energy entropy features. For the current study combinations of all these methods are used.

4.2 Problems Encountered in End-Point Detection Methods

End-point detection is very challenging field because of unfavourable conditions of real time recording. Most of the time SNR value is very low and noises are also unpredictable. It's difficult to find clean environment, where SNR is at least higher than the weak fricatives, so noise may not rise above them.

Speech data collected for current study are mainly recorded in an office environment. Common sources of noises are sound of another speaker, phone-ring, fan sound, etc. In some cases, speaker generated noises also create recognition problem. Such as coughing, heavy breath, less sound intensity.

Examples of the typical silence or background noise, for in-ear microphone recording and conventional recording system by keeping the microphone outside mouth is shown in fig 2 and 3 respectively, along with their estimated power spectral density.

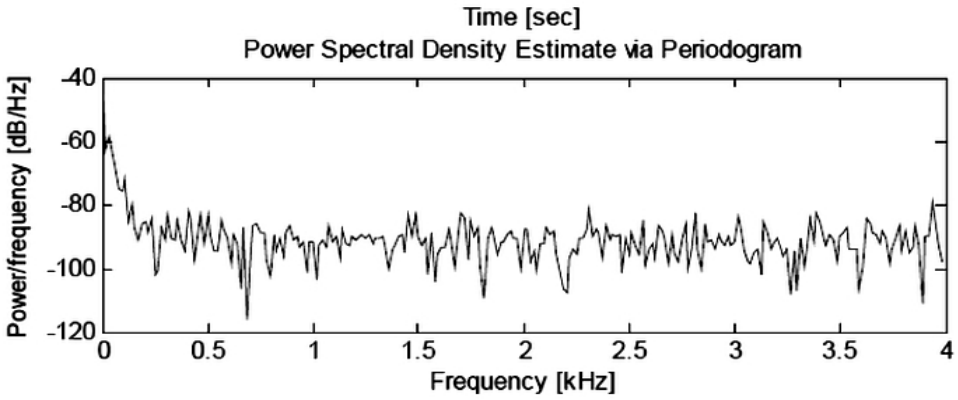


FIGURE 4.2: Typical Waveform For Recording Using In-Ear Microphone.

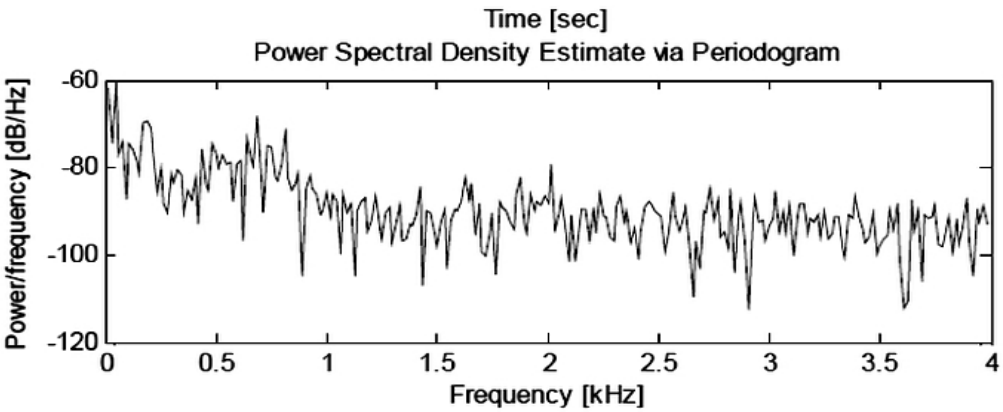


FIGURE 4.3: Typical Waveform For Recording By Keep Microphone Outside Mouth.

From the waveforms, we can observe that, in both kinds of recording silence portion includes low noise hum. In the case of in-ear recording low noise hum drop to -90 dB around 100 Hz. For recording using conventional recording system by keeping the microphone outside the mouth, hum drop below -90 dB at around 1.25 KHz. So, we can say that in-ear microphone structure creates low pass filter structure, and suppress high frequency noise hum.

The noise types encountered in real environments, which causes failure in the end point detection algorithms can be divided into three broad categories [Taboada, Feijoo, Balsa, Hernandz, 1994].:

1. Stationary noise associated with the transmission system, i.e microphone and/or surroundings.
2. Non-stationary noise, including people talking in the vicinity, door opening and closing, telephone ringing, mechanical (factory noise) and so on.
3. Noises and artifacts generated by the speaker such as mouth noises, sounds made by the tongue and lips, heavy breathing noises.

Last two noises are very difficult to deal with, and increases the complexity of the end point detection system. Non-stationary noises are long in duration and strong in intensity compared to the speaker. They can cover the full period of the speech duration and it can have very strong amplitude, compare to speech segment. A speaker generated artifacts are small in duration and can be detected as speech segment easily. In some isolated word recognition system, only one word may be spoken at a time. So speech recognition may consider speaker generated artifacts as a speech signal and start processing on it and may discard the actual speech segment.

Examples of different types of noises are shown in fig 4.4.and fig 4.5.

In both figure, band passed filtered speech waveforms are plotted on top, and their respective absolute energy waveforms are plotted at bottom.

Difficulties in the end point detections arise not only from the different types of noise present in the recording, but the types of the vocabulary words themselves also. Some phonemes or sound have very low energy when compared to the vowel portion of the speech, and they appear like background noises. Some examples of this kind of phonemes are: weak fricatives

at the start and end of the words, weak plosive bursts, final nasals, voiced fricative at the end of the word which become unvoiced, trailing off of the certain voiced sound.

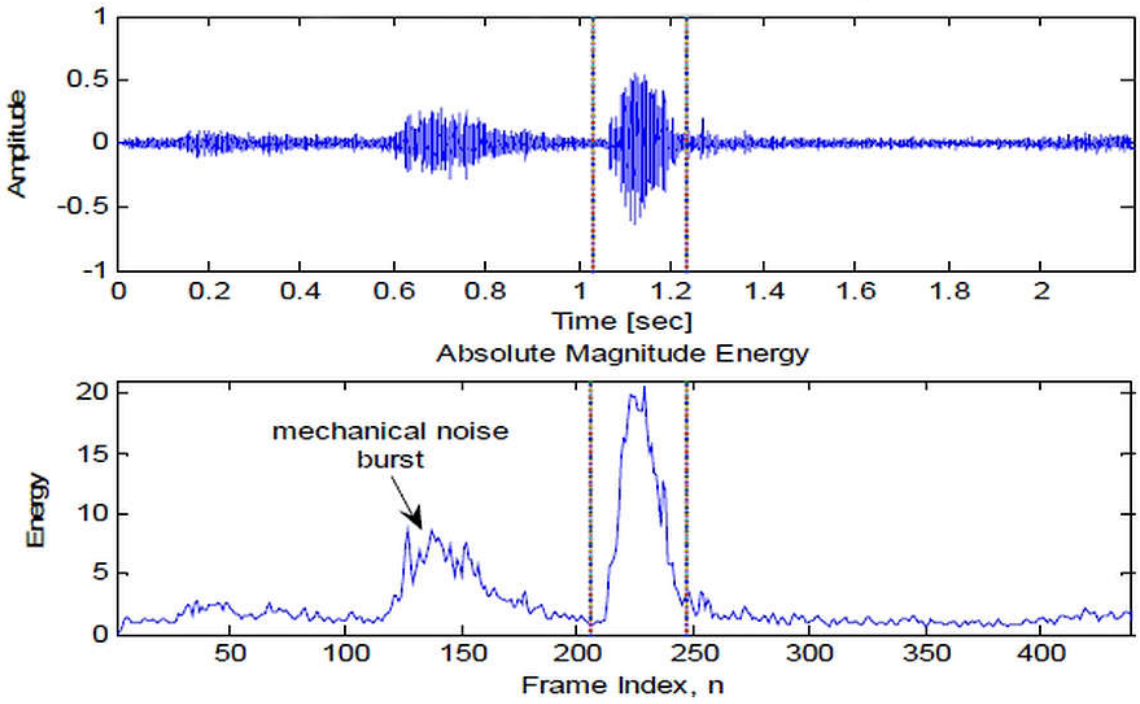


FIGURE 4 4: Mechanical Noises.

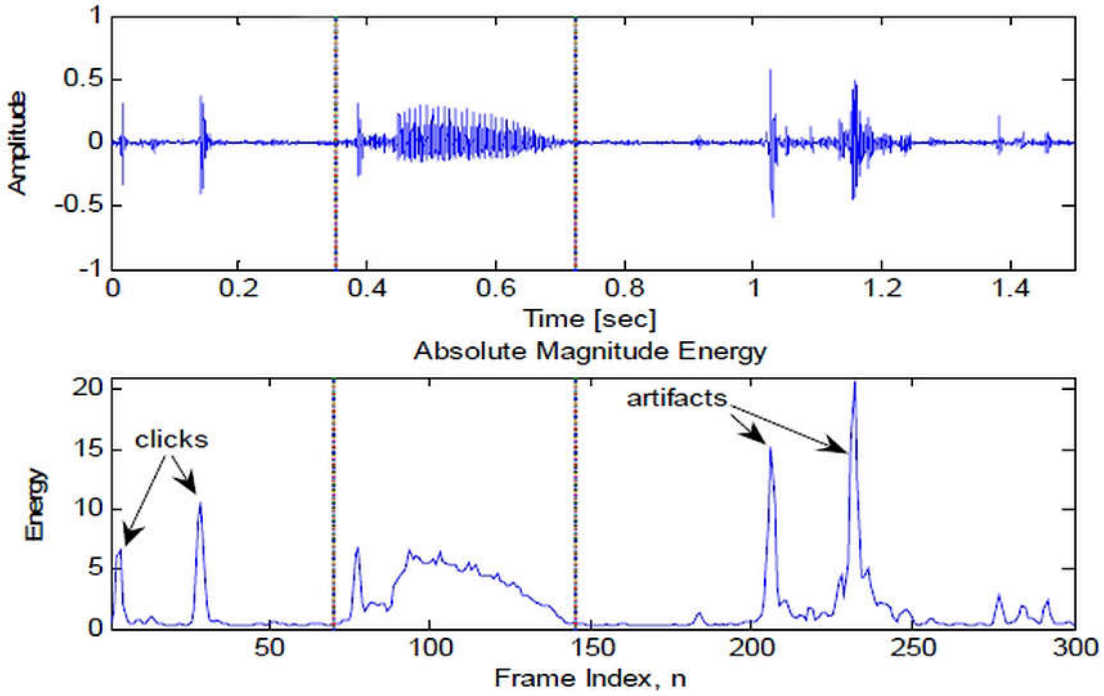


FIGURE 4.5: User Generated Noises.

4.3 End Point Detection Methods

This section presents some of the end point detection methods which are considered in this study. These measures include: short term energy, Teager's energy, short time zero crossing rate and energy entropy feature (EEF).

4.3.1 Short Time Energy Measures.

Short time energy measure has been used extensively in many end point detection system for finding initial word boundary. Because it's simple to use both in the software and hardware. Short term energy is the most natural way to represent the speech amplitude/energy variations.

For speech recognition system separation of the word from the silence portion is an important task, and then voice and unvoiced identification of the speech signal is the crucial part. Voice and unvoiced separation are required for the feature extraction methods. Voice signals are produced when the vocal cords vibrate during the pronunciation of the phoneme. Unvoiced signals by contrast produces sound output when air is exhaled from the lungs and not interrupted by the vocal cords. However, starting from glottis, somewhere along the length of the vocal tract, total or partial closure occurs which results into some user specific characteristic in unvoiced signal. Voiced signals as higher amplitude compares to unvoiced signal. So, short term energy can be used to separate voice from unvoiced signal. For extraction of speech from the silence portion signal to noise ratio should be higher than 30 dB, else lowest energy segment of speech signal will fall below noise available in silence portion [Rabiner, Sambur, 1975].

Short term energy parameters commonly used in end point detections are squared energy, logarithmic energy, root mean square energy (RMS), and absolute magnitude energy.

The squared energy parameter is defined as:

$$E_n = \sum_{i=1}^{N} x^2(i) \tag{4.1}$$

Where n is the frame index and N is the total number of samples in given frame.

The logarithmic energy is given by:

$$E(n) = \sum_{i=1}^{N} log_{10} x^2(i) \tag{4.2}$$

The root means square energy (RMS) is defined as

$$(n) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x^2(i)} \tag{4.3}$$

The absolute magnitude energy parameter is given by

$$E_n = \sum_{i=1}^{N} |x^2(i)| \tag{4.4}$$

The outputs for these four types of energy functions are shown in figure.

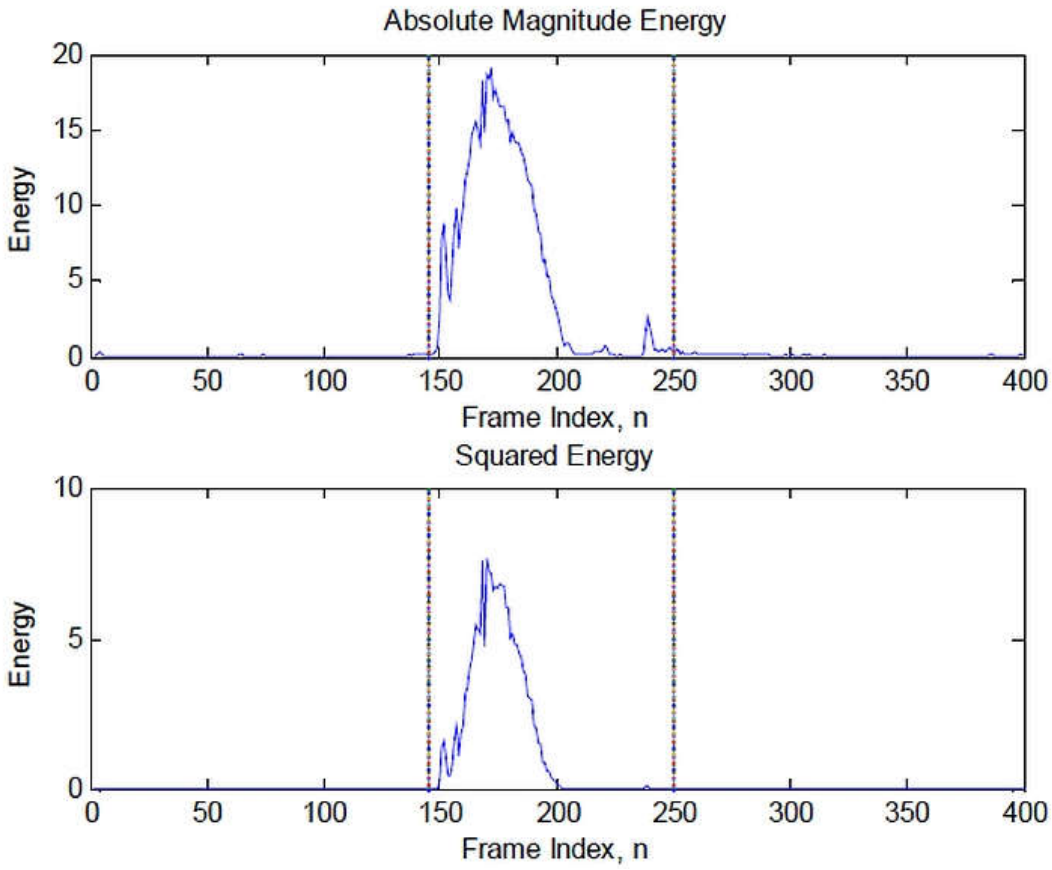


FIGURE 4.6: Absolute magnitude energy and Squared magnitude energy for word "tem"

The vertical dotted lines on each energy plot indicates the edge points of the utterance that are detected by the particular endpoint detection scheme. This is locally generated end points for that particular method only. By combining different end point detection methods, the final word boundary value will be applied.

The absolute magnitude energy reflects the sum of the magnitudes of sample amplitudes per frame, hence weak unvoiced segments of the utterance are not deemphasized, which contains useful information about speech segment. Its easiest method to detect the speech signal from silence portion. However, this method becomes unstable in week Signal to noise ration condition, as noise is not at all suppressed in this method. [Qiang, Youwei, 1998]

The squared energy method, suppress low noise completely. So, it will provide a better result compared to all other energy method scheme. But it also suppresses weak segments of the speech signals such as fricatives \f and stop consonant \t. Because of this disadvantage squared energy method can't be used standalone for end point detection scheme. It gives very conservative edge point estimates and mainly used to detect voiced portion of speech segments [Qiang, Youwei, 1998]

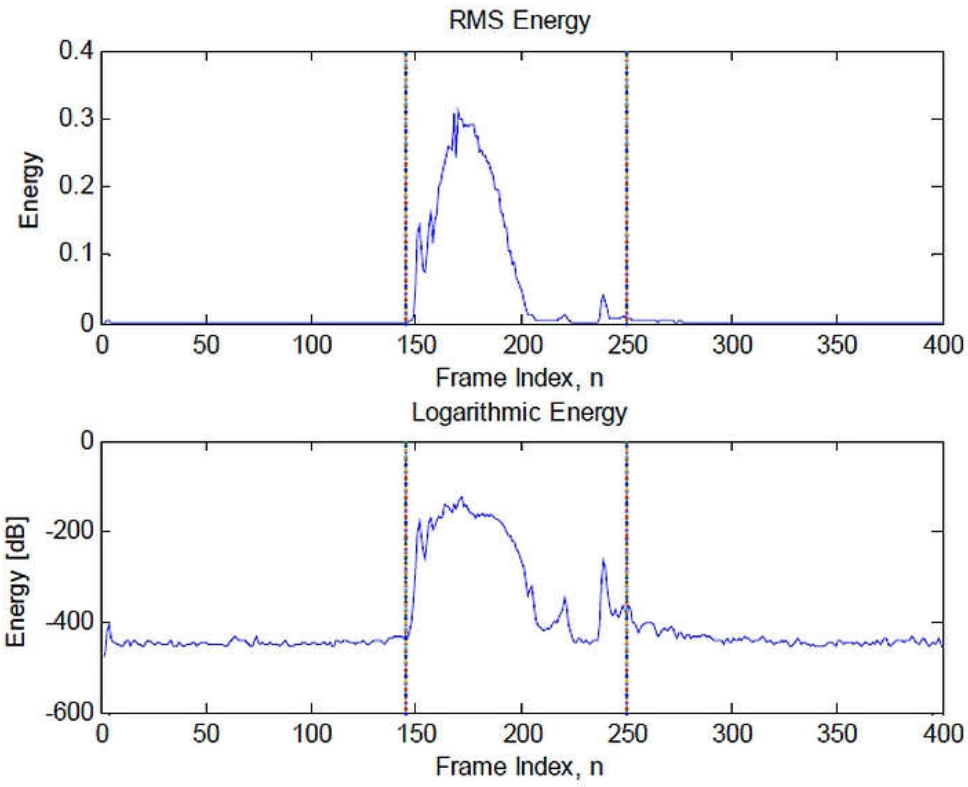


FIGURE 4.7: RMS energy and Logarithmic energy for word "tem"

The RMS energy resembles the squared energy in the sense that its scaled version of the squared energy parameter. The square root operator emphasizes low energy segments while deemphasizing higher energy signals, which makes it similar to absolute energy signal. So, it combines the advantages of the squared energy function and absolute energy function. But it increases computational load on the system.

Due to nonlinear compression property of the log function, the logarithmic energy reveals the details of the weak portions of the speech better than any other energy parameters. However, it also amplifies noise or silence portions of the recording and makes it harder to set threshold [Qiang, Youwei, 1998].

Above discussion highlighted the facts that, each energy function has its own advantages and disadvantages. Among all these energy based methods, the absolute magnitude energy method is simplest and fastest, making it suitable for online end point detection system.

4.3.2 Teager's Energy Algorithm

The Teager's energy algorithm, developed by Teager [Teager, 1980] for modelling of speech production. First time Kaiser [Kaiser, 1990] uses this algorithm to compute the energy of the speech signal. The basic idea behind the algorithm is as follows:

If the samples of the signal representing the oscillatory motion of the body are given by

$$x_i = Acos(\Omega i + \Phi), \tag{4.5}$$

where A is the sample amplitude,

Ω is the digital frequency in radian/sample

Φ is the arbitrary initial phase in radians

The energy of the signal is given as [Kaiser, 1990]:

$$E_i = x_i^2 - x_{i+1}x_{i-1} = A^2 sin^2(\Omega) \approx A^2\Omega^2 \tag{4.6}$$

The above equation is known as Teager's energy algorithm. Some observation can be made from the equation 6. Energy calculation is not only based on the single sample, but take past sample value also into consideration. So, instantaneous energy computed on time domain sample, captures dynamic variations of rapidly changing speech signal. The amplitude and

frequency variations of speech signals are also taken into account during energy measurement.

Above mentioned reasons suggest that, the Teager energy algorithm can be used as standalone feature for end-point detection algorithm. Ying et. Al [Ying, Mitchell, Jamieson, 1993] first implemented this algorithm for the end point detection. In place of calculating energy per sample basis, they calculated it per frame basis and they called resulted algorithm as "Frame based Teager energy" approach.

The power spectrum of the samples from the frame is first estimated from the Fast Fourier Transform (FFT) of the frame in the frame based Teager energy approach.

$$P_n(\omega) = \frac{1}{\frac{N}{2}} X_n(\omega).X_n^*(\omega) \qquad \text{For } \omega = 0 \ldots., \frac{\pi}{2} \qquad (4.7)$$

Where $X_n(\omega)$ is the FFT of the n-th frame. N is the total number of the FFT points in a frame, and ω is a digital frequency in rad/sample.

Now each sample in the power spectrum is weighted by the square of the corresponding digital frequency

$$P_n(\omega).\omega^2 \qquad \text{For } \omega = 0 \ldots., \frac{\pi}{2} \qquad (4.8)$$

Finally, frame energy is calculated by taking the square root of the sum of the weighted power spectrum defined in equation 4.8.

$$E_n = \left(\sum_{i=1}^{N/2} P_n(\omega).\omega^2 \right)^{1/2} \qquad (4.9)$$

Fig 4.8 shows the results of the Teager energy algorithm. From it, we can say that, the Teager's energy algorithm gives more conservative end point estimates then the short time absolute magnitude energy does. For online end point detection implementation, where the algorithm is forced to move from left to right for continuing updates of word boundary detection Teager energy algorithm require high computational task. The Teager energy algorithm also fails to capture exact end point locations when there is weak fricative and/or stop (plosive) consonants are present either in the beginning or end of the utterance.

The algorithm also not efficient and successful in tracking the changes in signals with multiple frequency components, and it is sensitive to noise when the signal contains different frequency component.

So, the Teager energy algorithm is not used in this work.

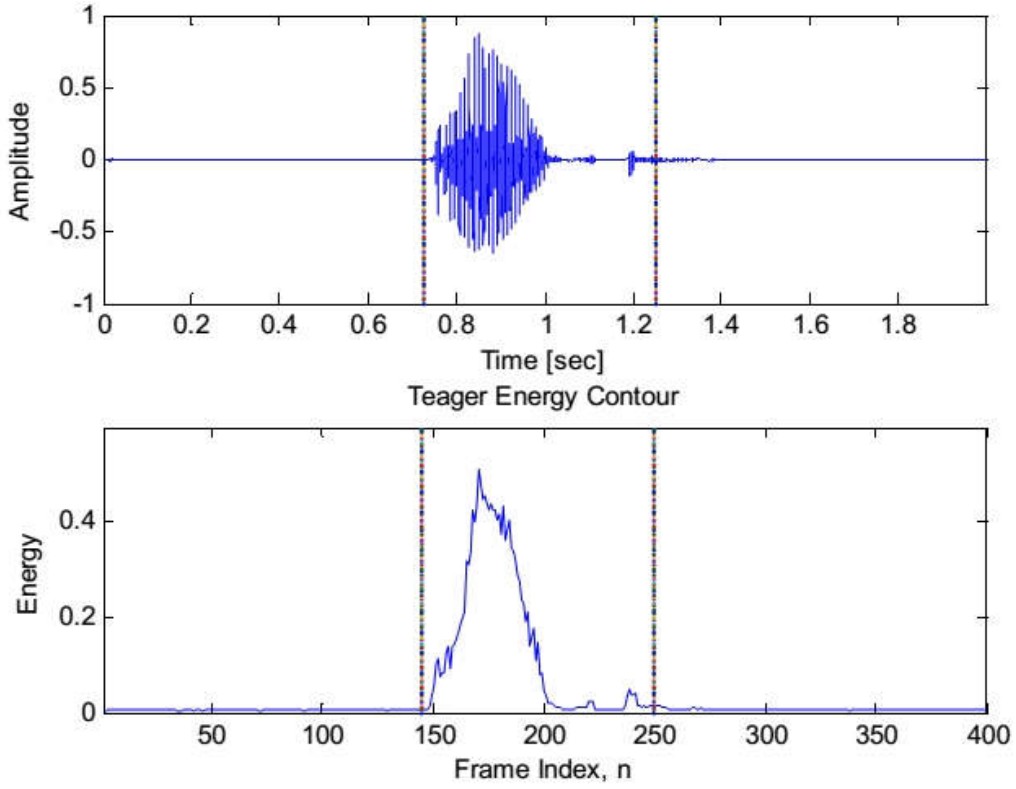


FIGURE 4.8: Speech Waveform and Teager Energy Plot.

4.3.3 Short Time Zero-crossing Rate (ZCR)

The Short Time Zero-crossing Rate (ZCR) is mostly used with short time energy, for end point detection. It's generally employed as secondary parameter to refine the initial end point estimates that were obtained by short term energy parameter.

The short term ZCR finds the number of times per frame, speech sequence changes its sign and given as:

$$(n) = \frac{1}{2}\sum_{m=1}^{N} |sgn[x(m+1) - sgn[x(m)]| \tag{4.10}$$

Where,

$$sgn[x(m)] = \begin{cases} +1, & x(m) \geq 0 \\ +1, & x(m) \geq 0 \end{cases} \tag{4.11}$$

The ZCR can give rough estimates of the frequency content of speech signals from the high SNR environment only. ZCR is very sensitive to DC offset, as it works on zero reference value. ZCR performance also affected by the low frequency hum. Last two problems can be solved by using DC offset removal and high pass filter prior to end point detection algorithm.

The performance of ZCR in noisy and less noisy condition is shown in fig. 9 and fig. 10.

The ZCR of the voiced speech segment is much lower and more steady than ZCR of silence segment. By looking to ZCR waveform it is not easy to set some threshold value to find the word boundary detection. ZCR plot is even worst for the noisy condition of speech signal.

4.3.4 Energy Entropy Feature.

The entropy is the measurement of uncertainty or the amount of unexpected information contained in signal, and its extensively used in the fields of coding and information theory. It was first applied to the end point detection problem for speech recognition by Shen [Shem, Hung, Lee, 1998]. All computational task is performed on a frequency domain, so it is also called as spectral entropy.

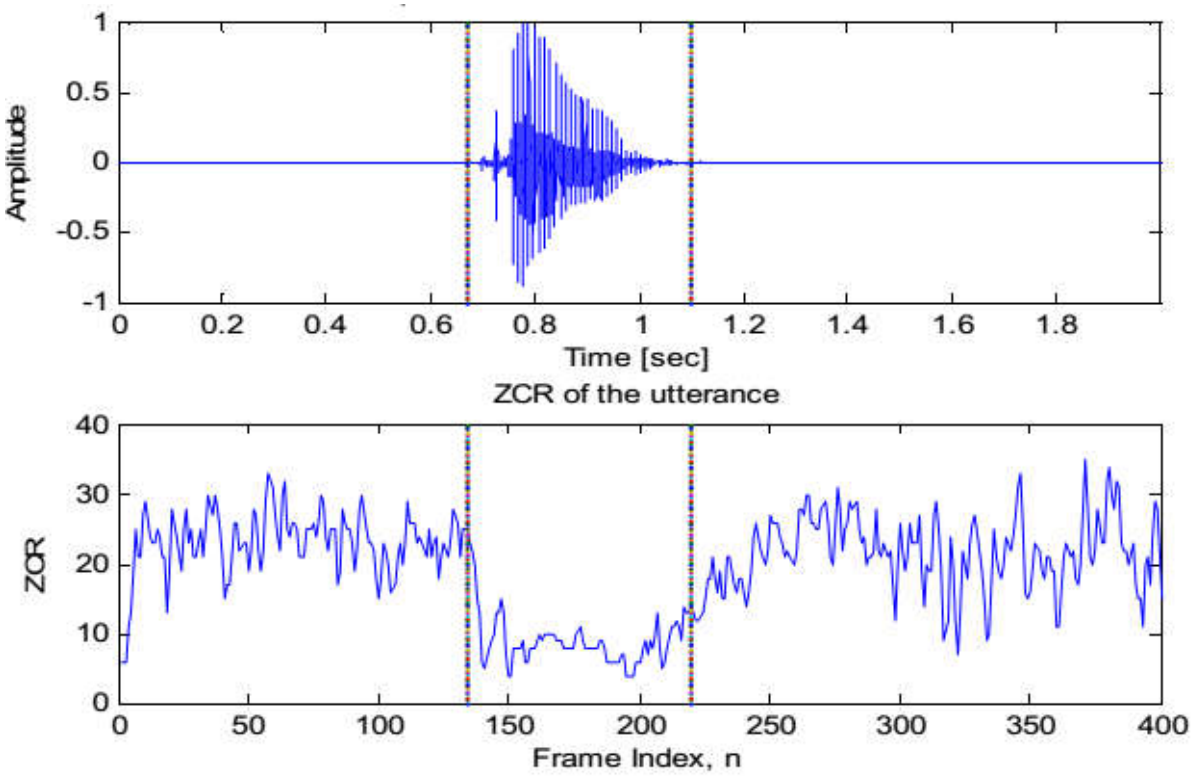


FIGURE 4.9: ZCR Plot for Noise Free Speech Signal.

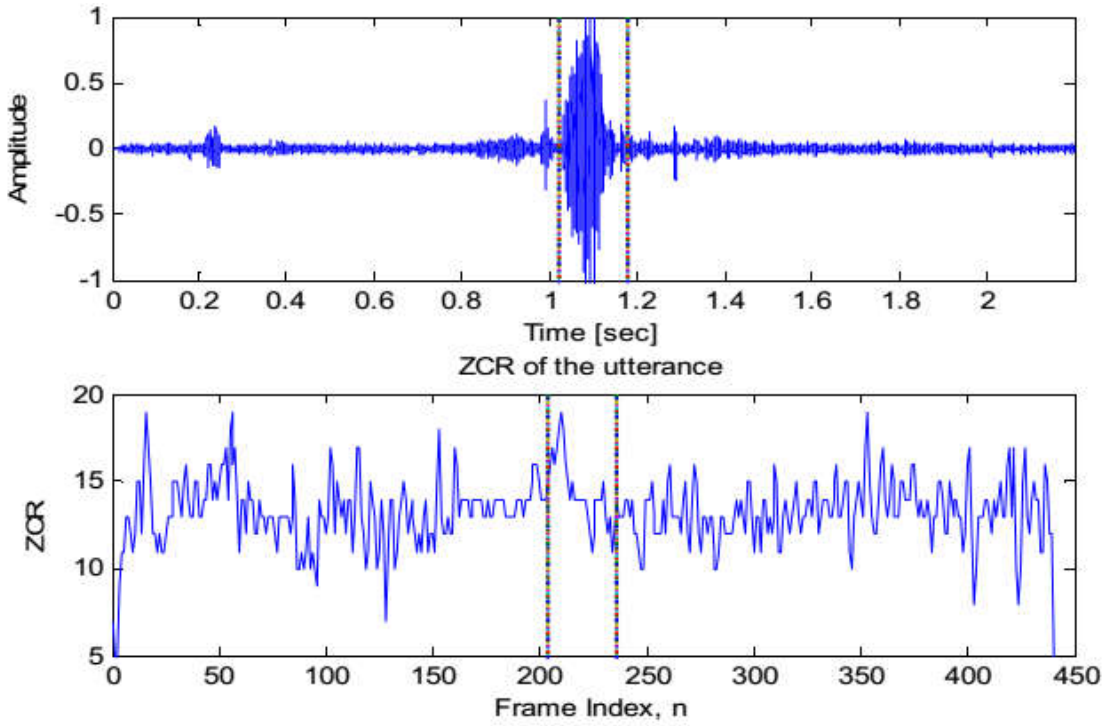


FIGURE 4.10: ZCR Plot for Noisy Speech Signal.

In order to obtain the entropy or spectral entropy of speech frame $X_n(m)$, first Fast Fourier Transform (FFT) of the frame is computed by using:

$$X_n(k) = \sum_{m=1}^{N} X_n(m) e^{-j2\pi km/N} \qquad \text{For k} = 1,2,3 \ldots \text{N} \qquad (4.12)$$

Second, spectral energy of frequency index k in each frame is estimated as:

$$S_n(k) = |X_n(k)|^2 \qquad \text{For k} = 1,2,3 \ldots \text{N}/2 \qquad (4.13)$$

Third, find the probability associated with each frequency index k, i.e the Probability Density Function (PDF), estimation of the spectrum, can be estimated by normalizing the spectral energies.

$$P_n = \frac{S_n(i)}{\sum_{k=1}^{N/2} S_n(k)} \qquad \text{for i=1,2,3...N/2} \qquad (4.14)$$

Fourth, band pass filter with pass band of [150 Hz, 2300 Hz] is applied to remove distortions present in all trials.

$$S_n(k) = 0 \qquad if\ f < 150\ Hz\ or\ f > 2300\ Hz \tag{4.15}$$

$$P_n(i) = 0, \qquad if\ P_n(i) \geq 0.9 \tag{4.16}$$

Finally, the entropy or the spectral entropy of a speech frame is defined as:

$$H_n = -\sum_{i=1}^{N/2} P_n(i).log_{10}[P_n(i)] \tag{4.17}$$

The entropy curve is shown in fig 4.11. Results obtained from it revealed that,

- Spectral entropy of speech segment is quite different from that of a silence segment.
- It performs well in the situation of the low SNR contaminated with different types of noises such as white noise, pink noise. But it fails under the multi-talker babble and background music.
- Short term energy quantity performs better in these noise conditions because of the additive property. Energy sum of speech+ noise will always be greater than energy of noise signal.

As we have observed that depending on the noise, in some situation short term energy performs well and in another situation entropy feature performs well. So, both have their own advantages and disadvantages. So, it is advantageous to combine both these methods in one algorithm to have more robust and reliable end point detection algorithm. The name given to this combined method is an Energy Entropy Feature.

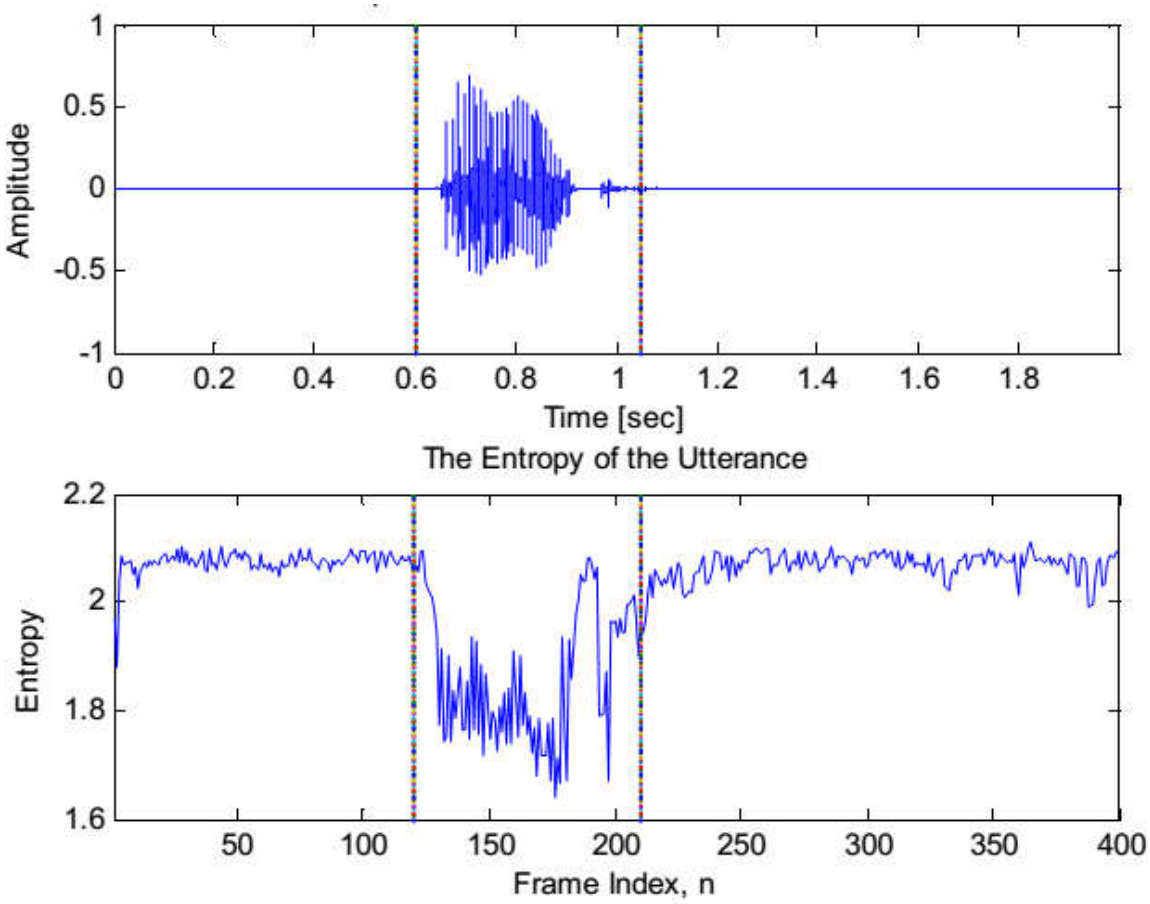


FIGURE 4.11: Entropy Feature Curve.

Energy Entropy Feature (EEF) is formed by simply multiplying the energy and the entropy computed for the specific frames. It is given by:

$$EEF_n = \sqrt{1 + |E_n \times H_n|} \tag{4.18}$$

The multiplication operation in EEF, emphasizes the voiced regions and attenuates silence or noisy regions. It also emphasizes Low energy areas such as weak fricatives and stop consonants at the end or at the front of an utterance. So, even in the low SNR value, EEF can be used for refinement of end-point detection done by short time energy.

The energy entropy (EEF) waveform along with absolute magnitude energy for comparison is shown in fig 4.12.

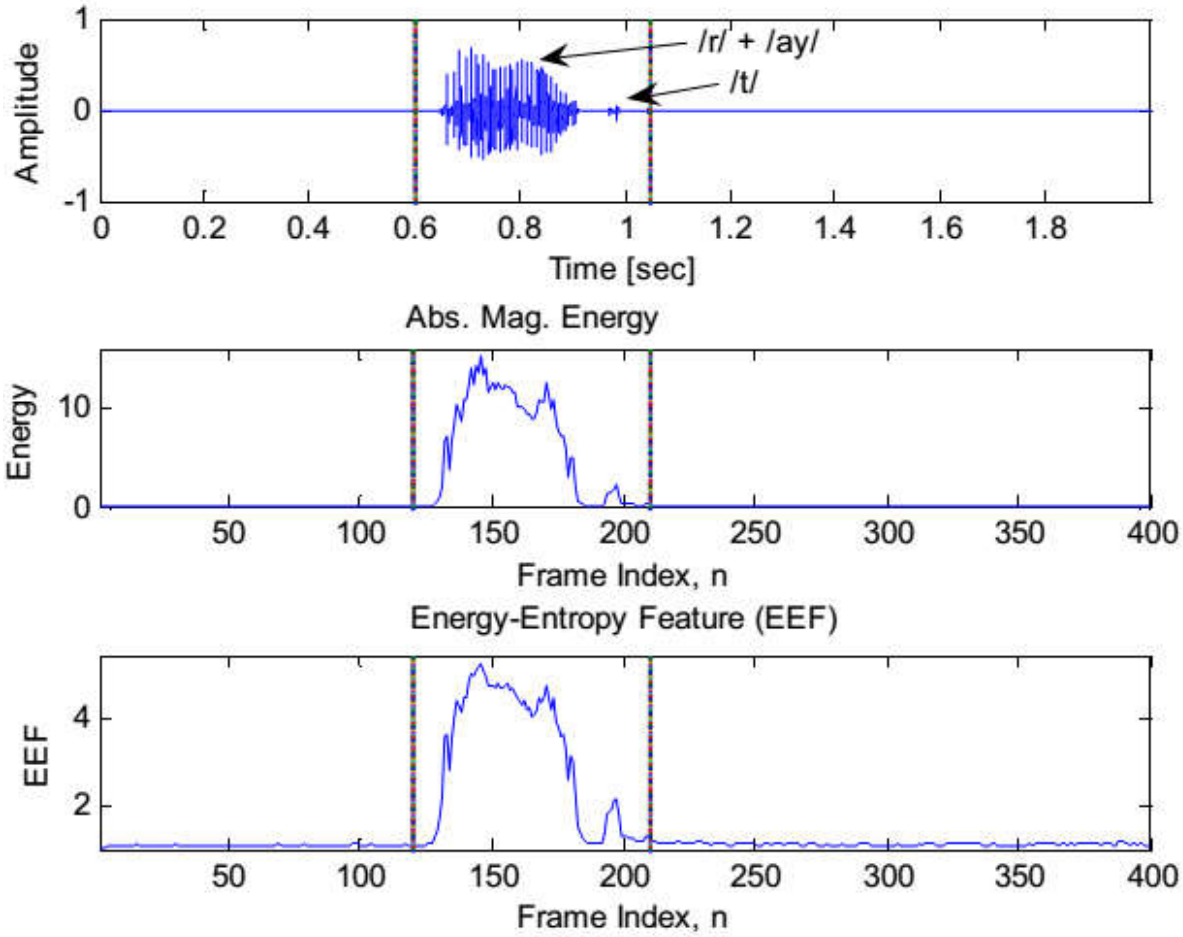


FIGURE 4.12: Absolute Magnitude Energy and Energy Entropy Curve.

Fig 4.12 shows that, magnitude difference between vowel portion and consonant portion is reduced. So, low-energy stop consonant \t following the diphthong \ay is more emphasized and visually noticeable in the EEF plot compared to energy or entropy plots. It also emphasized the background noise sections compared to energy functions of utterance. The solution for this problem is to use different short term energy quantity for the EEF, namely short time squared energy quantity.

Due to squaring operation, short time squared energy quantities provide greater attenuations of the background noise than the short time absolute magnitude energy quantity. Short time squared energy and short time absolute energy not used together because using different parameter for the EEF would mean additional computation and reduced speed of the detection algorithm. The threshold value for the squared energy algorithm used with EEF, is calculated based on the silence sections.

4.4 End Point Detection Algorithm

This section presents the end point detection system used for present study. First, we discuss pre-processing stage used before end point detection algorithm. Next framing issues are discussed. Then threshold methods used in this algorithm are discussed. Finally, overall end point detection system is explained.

4.4.1 Pre-processing Stage

As discuss earlier, there is a low frequency hum in the frequency interval [0,100 Hz] present in all recordings, and the frequency content of the recorded utterance mainly concentrated up to 2.3 KHz, for in-ear microphone recording. It also exhibited strong stationary noise around 2.4 KHz.

To solve this problem, recordings are passed through the IIR band pass filter having the following characteristics:

- Elliptical IIR filter
- 9^{th} order
- Passband region 150 Hz to 2.3 KHz.
- Stopband region [0 Hz, 100 Hz] and [2.35 KHz, 4 KHz]
- At least 50 dB attenuation in the stop band

An IIR filter preferred over the IIR filter because required specification can be achieved with the lower order implementation so increases processing speed. For comparison, one stage IIR filter with same specification is equivalent to FIR filter of 300^{th} order, which is not suitable for the real-world applications.

4.4.2 Framing

The speech, signals is non-stationary in nature. For a short time, duration its assumed to be a stationary and periodic. So, speech signals are divided into the frames of 10 ms. Different windowing methods can be used for framing. Windowing method other than rectangular one may complicate the computations and the frame shift may produce variable weight in each frames. In the current work rectangular window is used, as it is easiest one to use in on-line implementation and it weights all the samples in frame equally. The speech recognition system application for the current work is mainly operated on the time domain basis so,

spectral resolution is not the issue in the current application and we can use any type of windowing method.

The frame length is considered to be 10 ms. The all speech signals are recorded with sampling frequency of 8 KHz so, there are 80 samples per 10ms long frame. The frame rate is 5 ms, i.e. 50 % overlapping is used between the frames. Overlapping allows the capturing the dynamic variations from one frame to another and increases the accuracy of the end point detection as, end point detection is made on frame basis.

Note that, there is no actual multiplication performed to form the frames since rectangular window is used for framing. Sliding window with overlapping of 50 % moves from left to right, which takes 10 ms long frames or 80 samples per frames of the incoming speech signal is applied. So, there is not required to store the speech signal in matrix prior to detection. This solves the requirement of the online implementation of end point detection algorithm.

4.4.3 Threshold Mechanism

Speech end point detection algorithm accuracy is highly depending on the threshold mechanism system. Algorithm often failed to detect speech segment and overall speech recognition system fails if we use only one threshold mechanism. So, in current work two thresholds, Upper and lower threshold are computed and supplied to the energy based detection phase. Another advantage of two level threshold is: upper threshold level speed up the algorithm since upper threshold is much higher than maximum energy of the most of the user generated artefacts and sporadic noises. So, the algorithm does not spend much of the time in the noisy section of the speech signal. The lower threshold is used to refine the end point estimates obtained with the upper threshold value, since upper threshold is more conservative than the lower threshold to avoid sudden noise burst that are strong in energy.

The silence portion of the recorded speech signal is used to calculate the stationary noise present in the speech signal. Based on its values upper and lowers threshold values are calculated. In current work assumption is made that initial 100 ms of the recording are silence portion. The silence portion can be refined using short term energy calculation and zero crossing detector system. First 100 ms period means 20 overlapping frames. The silence energy $E_{silemce}$ is computed by taking the average of the short term energy for the first 20 frames. Upper threshold ITU and Lower threshold ITL, are then obtained by multiplying $E_{silence}$ with scalar quantity based on the $E_{silence}$ value. Therefore, ITU and ITL are updated on a real time basis.

Note that it's not always true that first 100 ms of recorded signals are silence portion, in some cases speech may start right at the beginning of the speech recorded started. In this kind of situation $E_{silence}$ may become quite large, which can create detection errors. As a solution of this problem, if $E_{silence}$ value becomes greater than predetermined level 4.5, then its value will be reset to 0.5 and threshold are calculated based on the modified reset value. These two values are calculated based on the trial and error method. After use of this values, chances missing entire utterance, as it can be normally happened for energy based algorithm will be almost zero.

Sometimes it may possible that first 20 ms frames contained the strong user generated noise or sporadic noise which makes silence portion difficult to identify. Compare to stationary noise this noise is of short duration and high in amplitude. So it makes difficult to identify the actual silence portion of the reference. In some cases, it might be happened that more than first 20ms period is totally absent because of recording started late. As a solution of these problems, for better estimates of $E_{silence}$ algorithm start searching for the right direction until $E_{silence}$ drops below predefined value of 2.0, or until reaching predetermined level of frames, i.e. 70 frames, whichever comes first.

4.4.4 End point detection algorithm

In this section, actual algorithm used for the end point detection is discussed. The end point detection algorithm designed for this study uses two parameters as discussed earlier, namely the short time absolute magnitude energy and energy entropy feature (EEF). The beginning and end points of the utterance are calculated in two steps. First, short time absolute magnitude energy quantity is used to obtain the initial edge point estimates of the speech signal. Second refinement of the initial end point estimates is performed by the energy-entropy feature (EEF). After this, end points are declared and speech segment of the recording corresponding to the section between these end points is cropped and forwarded to the feature extraction block prior to classification and recognition stage.

Most of the information used by detection algorithm is obtained from the short time absolute magnitude energy steps. To increase the reliability and robustness of the detection algorithm some heuristic assumptions are made and involved, such as duration and maximum energy value.

Some non-stationary noises present in the speech recordings such as artefacts generated by the speakers, exceeded the upper threshold and were erroneously classified as speech.

Mostly they are short in duration and produced due to clicks, pops and lip smacks and they are strong in energy. In such cases, duration count can help to solve problem of false detection of noises as speech signal. Duration count used in these algorithm is 10 frames or 50 ms, and it was selected empirically. Basically, duration of the voiced section of the shortest word in the vocabulary "અામ" is around 20 frames or 100 ms on average. However, using 20 frames as the duration count value directly resulted in lots of detection errors such as, it misses or false detection with word "અામ".

The algorithm starts to count the number of frames that are above the upper threshold once the energy of a frame exceeds the upper threshold. Speech is declared to be started when number of frames exceeding the upper threshold is 10 (or 50 ms in duration)

There are also few cases where the algorithm fails to detect the utterance or sometimes consider noise as utterance. This may be considered as false rejection and false acceptance. This may happen due to following reasons.

- Strong spikes of short duration, generated by the user just before the beginning of the voiced region, such as in the case of word "ઉપર".
- High energy noises such as mechanical noises or strong background voices, which are non-stationary in the nature and present for the full duration of the word.

As mentioned, due to the first reason, energy based scheme may miss full utterance for some word like "ઉપર". It means even though word is present, but it misses it so it's a case of false rejection. As a solution of this problem, if energy algorithm not able to find any speech utterance in given speech signal then energy contour applied to 5^{th} order median filter to push the energy level below upper threshold value and it will start searching again from the beginning of recording. This process continues till it's not able to find any speech utterance for given recording samples.

The second reason, causes the energy algorithm to detect noise as speech signal. It means its case of false acceptance. As a solution of this problem, after finding the presence of the word, it passed through the 5^{th} order median filter to smooth its energy value. The energy of the detected signal, then compare with the energy of the full signal. Results show that maximum energy of the speech segment is always higher than the energy of the non-speech segments. This condition, if not satisfied, then it again starts searching for correct speech utterance.

Modified energy algorithm performs well in most of the condition, but still there are some cases where speech end point detection can't rely totally on energy scheme for word boundary detection. So, along with the energy scheme, entropy calculation is done and combined method called Energy Entropy Feature method (EEF) is suggested.

Two threshold levels are used for Energy Entropy Feature method (EEF) method. The threshold levels are not fixed or static values, but its values are calculated from the silence portion of the recorded signal. This logic helps us to provide dynamic threshold level, and also able to remove constant stationary noises. To remove the noise from speech signal, we applied logic as follows: if constant noises such as fan sound or some fixed background noises, which remain almost constant throughout the full speech signal then it's easy to find out its strength. Because during initial silence portion whatever energy is present that is because of this noises only so, same energy will be subtracted from the total speech signal so remaining energy will be due to spoken word or utterance only. The energy present in the silence portion is called as $E_{silence}$.

Two threshold mechanism used to fasten the speech end point detection process. First end point used to refine speech start point while the second is used to refine the end point only. These end points values are derived from the value of $E_{silence}$. As there are no fricatives at the starting of the word and energy is also high, so it's comparatively easy to find the word starting point. So if starting point is missed to find then EEF searches only 5 frames again from the initial value, but as end point is difficult to find so EEF again searches in 70 frames to detect missed end point.

The overall end point detection method can be summarized as follows:

- Select initial threshold values ITU and ITL using threshold mechanism explained earlier
- Calculate short time absolute magnitude energy for the first selected frame and define it as $E_{n.}$
- Compare value of $E_{n.}$ with the ITU
- If the ITU $\geq E_{n.,}$ continue search with next frame.
- If $E_{n.} >$ ITU then check short time absolute magnitude energies of the next ten frames. i.e next 50ms, (duration count). If all 10 frame energies are above ITU, declare first

frame, where $E_{n.}$ > ITU as the preliminary beginning point of the utterance $\overline{SP}$. Otherwise continue the search.

- After the preliminary beginning point $\overline{SP}$ is found with ITU, refine it by searching back to the first point at which $E_{n.}$ > ITU is found. Label it as the speech start point, SP.
- Continue forward until $E_{n.}$ < ITL to find possible end point locations.
- If $E_{n.}$ < ITU, then check the energy of the next 10 frames, in order to avoid sudden energy dip. If the short time energies of these 10 frames are not all above ITU, declare the first frame whose $E_{n.}$ < ITU as the preliminary end point of the utterance $\overline{EP}$. Otherwise continue the search.
- After the primary end point $\overline{EP}$ is determined with ITU, refine it by searching forward until the first point at which $E_{n.}$ > ITU is found. Label it as the "speech end point", EP.
- Check the index of the speech start point. If it is found to be equal to the last frame index of the whole recording N, then the algorithm has completed a right to left run on the whole recording without finding any start and end point. Thus, a miss had occurred.
- If the speech starts and end points are found to be the last frame index and zero respectively, then a miss has occurred.
- In the case of miss, we smooth the short time absolute magnitude energy curve with 5^{th} order median filter, and conduct the same search procedure explained above on the smoothed energy curve from right to left, once again to find the speech end points, SP and EP.
- If there are no misses, then compute the energy of the rest of the recording, and smooth it with 5^{th} order median filter.
- Find the maximum of the smoothed energy curve E_{median}, of the whole recording $\max(E_{median})$, and maximum of the smoothed energy curve between the detected end points $\max [E_{median}(SP: EP)]$.

- If max [E_{median}(SP: EP)] = max(E_{midian}), then terminate the energy algorithm, and returned the edge point estimates that will be used by the EEF algorithm to refine the estimates if necessary.
- Otherwise performs the search procedure outlined above on the E_{median} until max [E_{median}(SP: EP)] = max(E_{midian}).
- Terminates the energy algorithm, and return the edge point estimates for final refinement by the EEF algorithm.
- Call the EEF algorithm to refine the start/end points.
- Set the thresholds, $EETL_1$ for refining SP, and $EETL_2$ for refining EP, according to $EEF_{silence}$.
- Search five frames to the left starting from SP. If EEF_n>$EETL_2$ (n is between SP-5 and SP-1), move the SP to that end point. Otherwise, do not change the initial start point SP. Declare the "final speech start points", FSP.
- Search 70 frames to the right starting from EP. If EEF_n>$EETL_2$ (n is between EP+1 and EP+60) check the EEF of the next 10 frames (to take into account the duration of the stop consonants.), and move EP to the last frame at the far right where EEFn>EETL2. Otherwise, do not change the initial end point EP. Declare the "final speech end point", FEP.
- Complete the end point detection algorithm and crop the speech signal from the estimated start/end points, FSP and FEP.

The results of end point detection methods suggest that its very effective and having zero error. Means for all speech samples given to end point detection algorithm able to find word boundary for it. They never return misses, means error massage of "no word boundary detected". Word boundary is detected by combining different methods so, results are very accurate too. They never detected noise signal as speech, and never detect noise signal as speech segment.

After detection of end point using suggested algorithm, speech segment is passed through the filter section. As stated earlier, 9^{th} order elliptical IIR filter is used. IIR filter is used here because it's easy to implement in real time as it required less number of coefficients compare to the FIR filter.

If we want to design similar type of filter using FIR, then design parameter will be as follows:

- Bandpass equiripple FIR filter
- 300^{th} order
- Pass band from 150 Hz to 2.3 KHz
- Stopband is [0 Hz, 100 Hz] and [2.35 KHz, 4 KHz]
- At least 50 db attenuation in stop band

The results of FIR and IIR filters are shown in figure. Waveform shows that, IIR filter behaves abnormally in the case where there is a heavy breath taken by speaker either at starting or at the end of the word. IIR filter amplifies this portion and shift word boundary further right or left respectively. Due to this, its unable to find exact end points of speech signal.

FIR filter is more stable compare to IIR so, it's not amplifying unwanted breath release of the speaker. It provides more accurate end point detection compare to IIR. Main drawback of the FIR filter implementation is the higher order requirement which makes it unsuitable for the real time applications. However, with help of multi-rate signal processing techniques, we can achieve it with reduced overall complexity of the system [Mitra, 2006].

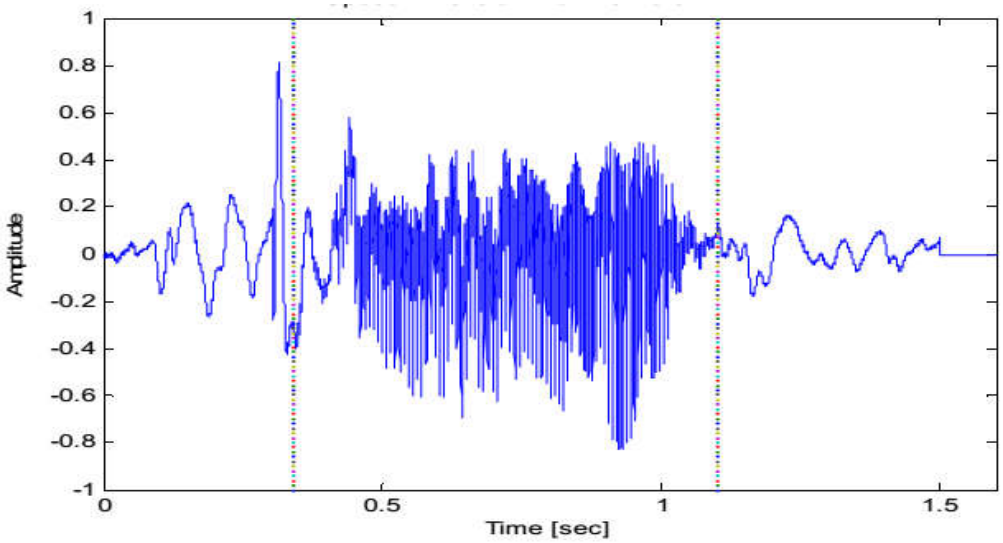


FIGURE 4.13: speech waveform for word "અમે"

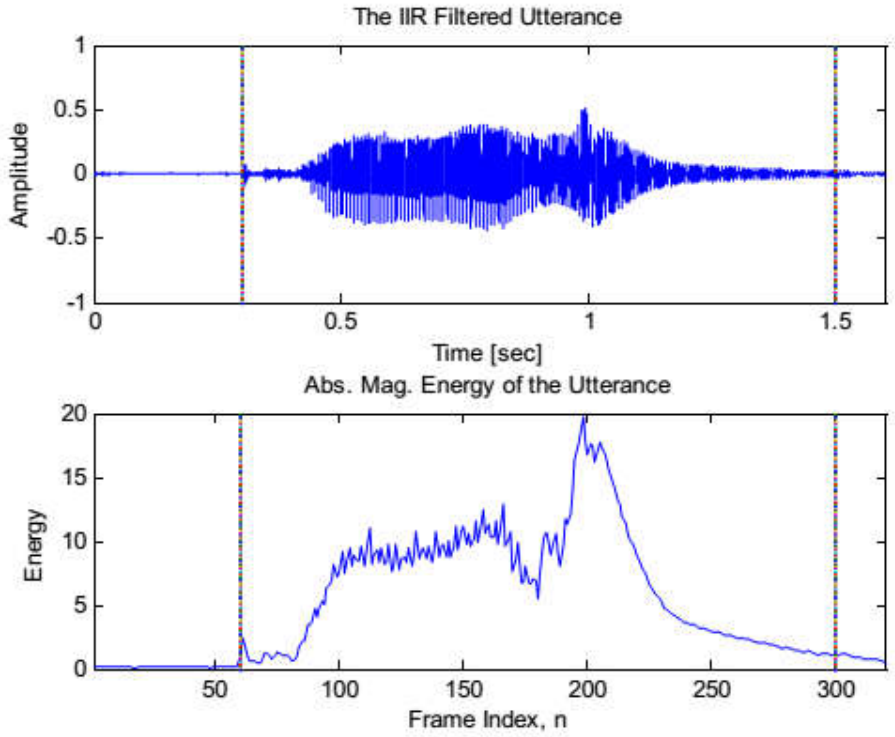


FIGURE 4.14: IIR filtered utterance for word "આમ" and corresponding absolute magnitude energy. Detected word boundary is indicated by the dotted line.

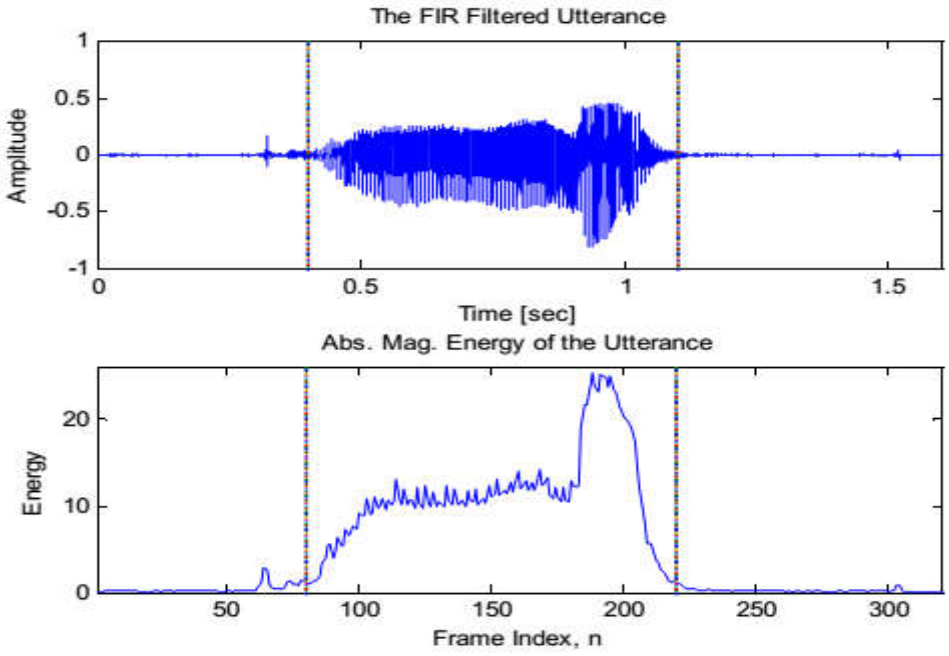


FIGURE 4.15: FIR filtered utterance for word "આમ" and corresponding absolute magnitude energy. Detected word boundary is indicated by the dotted line.

4.5 Summary

This chapter discussed about the

- Importance of the end point detection system
- Types of end point detection methods
- Different factors affecting the end point detection methods
- Types of stationary and non-stationary noises.
- Methods to minimize the effect of noises
- Different energy based schemes
- Use of entropy methods
- Combinations of energy and entropy features.
- Steps for word boundary detection.
- Use of FIR and IIR filter.

CHAPTER 5

Feature Extraction

This chapter presents different feature extraction methods used for the speech recognition. Feature extraction provides compact representation of the segmented speech data to be used at recognition stage. Mainly two types of spectral feature extraction methods are discussed, namely Real Cepstrum (RC) coefficient, and Mel-Frequency Cepstral Coefficients (MFCC).

5.1 Real Cepstrum (RC) Coefficient

Speech production is usually modelled as the convolutions of an excitation sequence e(n) with the impulse response of the vocal tract, h(n). Therefore, the speech x(n) can be represented as:

$$S(n) = e(n) * h(n) \tag{5.1}$$

For the frequency domain representation, equation one is represented as multiplication of Discrete-Time Fourier Transform (DTFT) of two sequences:

$$S(\omega) = E(\omega) * H(\omega) \tag{5.2}$$

The excitation sequence is considered to be a random noise sequence for unvoiced speech and a quasi-periodic impulse train with the pitch period for voiced speech, whereas impulse response of the vocal tract is considered to be a short window. Therefore, exciting sequence is viewed as the rapidly varying part of the speech, as opposed to the vocal tract filter which represent the slowly varying part of the speech.

In many speech application, separate estimation of the excitation sequence and the vocal tract model is required. Since the speech is produced by convolution operation in time

domain or multiplication in frequency domain. It's difficult to separate the speech into two sequence by well-known linear techniques. As applied first by Noll [Noll, 1967], the excitation and vocal tract model can be separated by using a nonlinear operator, namely, by taking the logarithm of the signal:

$$log|S(\omega)| = log|E(\omega)| + log|H(\omega)| \tag{5.3}$$

Cepstral analysis is motivated by the idea of separating these two components of the speech signal. It was first discovered and applied to seismic analysis by Bogert [Bogert, Healy, Tukey, 1963], but the cepstral analysis was extended and first applied to speech by Noll [Noll, 1967], [Deller, Proakis, Hansen, 1993]. The more general mathematical model which is called homomorphic (cepstral) signal processing was introduced by Oppenheim [Oppenheim, 1969]. The cepstrum derived from the homomorphic processing is usually known as the complex cepstrum, although its version used in practice is called the real cepstrum [Deller, Proakis, Hansen, 1993].

The main difference between the real cepstrum and the complex cepstrum is that the phase information about the speech signal is retained in the latter while it is discarded by the real cepstrum [Deller, Proakis, Hansen, 1993]. Even though the complex cepstrum might seem to be more attractive since it preserves more information than the real cepstrum does, it is not used much in practical applications unless it is desired to return back to the time domain [Gold, Morgan, 2000]. Real cepstrum coefficients are more commonly used in speech recognition applications as there is no return to the time domain once features are extracted.

The real cepstrum (RC) of a speech sequence s (n) is defined as the inverse DTFT of the logarithm of the spectral (DTFT) magnitude:

$$c_s(n) = IDTFT\{log|DTFT\{s(n)\}|\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} log|S(\omega)|e^{j\omega n}d\omega \tag{5.4}$$

where the natural or base 10 logarithms is generally used in the definition. Since the speech sequence s (n) is real-valued, its logarithmic spectral magnitude, log |S (ω)|, is real and even. Therefore, the real cepstrum $c_s(n)$ is real and even, i.e., the second half of the real cepstrum coefficients is redundant and repetitive of the first half. The computation process of the real cepstrum is shown as a block diagram in Fig 5.1.

In practical applications, the DTFT and IDTFT in the above definition are replaced by the discrete Fourier transform (DFT) and inverse discrete Fourier transform (IDFT), respectively. In this case, Equation (5.4) can be modified as:

$$C_d = \frac{1}{N}\sum_{i=0}^{N-1} log|S(k)|e^{j2\pi kn/N} \qquad \text{for } n = 0, 1..., N-1 \qquad (5.5)$$

Since the speech is time-varying, it is blocked into small frames where it is assumed to be stationary, thereby allowing to capture speech temporal and spectral dynamic changes. Hence, the above real cepstrum definition is also known as the short time real cepstrum since it is applied to each individual frame.

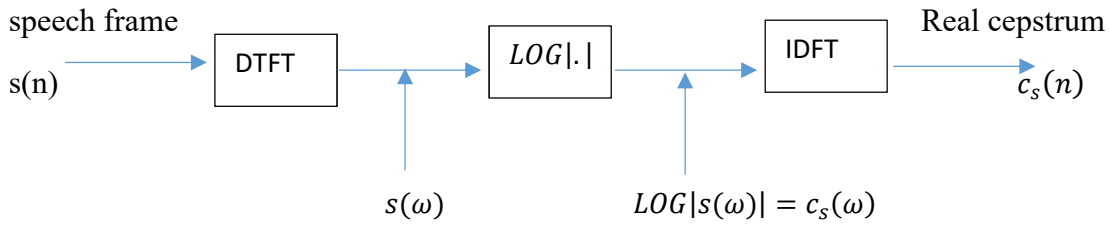


FIGURE 5.1: Computation of the real cepstrum using the DTFT (After: [Deller, Proakis, Hansen, 1993]).

The real cepstrum computation in Fig 5.1 is modified according to Equation (5.5) and shown in Fig 5.2.

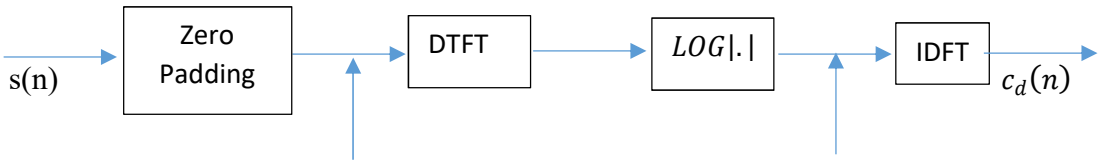


FIGURE 5.2: Computation of the real cepstrum using the DFT (After: [Deller, Proakis, Hansen, 1993]).

Using the DFT instead of the DTFT is similar to sampling the DTFT at *N* equally spaced frequencies from -*n* to n. In this case, c_d (*n*) in Equation (5.5) becomes the convolution of the original c_s (*n*) in Equation (5.4) with a uniform impulse train of period *N* [Deng, O'Shaughnessy, 2003]:

Therefore, the real cepstrum computed with the DFT and IDFT contains the replicas of the real cepstrum computed with the DTFT and the IDTFT at intervals of N. This in turn may potentially cause some aliasing, but can be minimized when the number of DFT points *N* is kept large enough, typically more than 100 points [Deng, O'Shaughnessy, 2003]. Thus avoiding aliasing can be achieved by either selecting longer frames or zero padding, as indicated in Fig 5 2.

For speech recognition purposes, the number of real cepstrum coefficients retained is generally less than 20 and typically 10 to 14. This ensures that the speech portion due to the vocal tract is kept while removing the contribution due to the excitation. Hence, for the current study, 14 real cepstrum coefficients are used for each speech segment cropped by the endpoint detection algorithm discussed in Chapter 4.

The real cepstrum coefficients derived from one of the segmented utterances of the word "tem" are plotted in Fig 5.3.

5.2 Mel-Frequency Cepstral Coefficients (MFCC)

The linear predictive coding (LPC) and real cepstrum coefficients were the major parameters used to represent utterances for speech recognizers up until the 1980s [Deller, Proakis, Hansen, 1993]. The mel-frequency cepstral coefficients (MFCC) were first used for a speech recognition system with a dynamic-time warping algorithm (DTW) in a study by Davis and Mermelstein in 1980 [Davis, Mermelstein, 1980]. Their study revealed the fact that MFCCs outperform any other parametric representation such as LPC and real cepstrum coefficients. MFCCs developed by Davis and Mermelstein [Davis, Mermelstein, 1980] have become the most popular features up to date.

Real Cepstrum Coefficients of the Word "tem"

The basic idea behind using MFCCs is to obtain a feature representation which approximates the human perception. MFCCs are the nonlinearly weighted and warped (with a nonlinear mapping scale) versions of the real cepstrum coefficients [Deng, O'Shaughnessy, 2003]. Basically, the logarithmic spectral magnitudes on the physical frequency scale obtained by the real cepstrum are warped to corresponding magnitudes on a perceptual frequency scale since the perceived pitch or frequency of a tone is different than the physical frequency in Hz [Deller, Proakis, Hansen, 1993]. As a result, the mel scale allows for the required

mapping (or warping) from the physical frequencies to the actual perceptual frequencies, as shown in Fig 5.4.

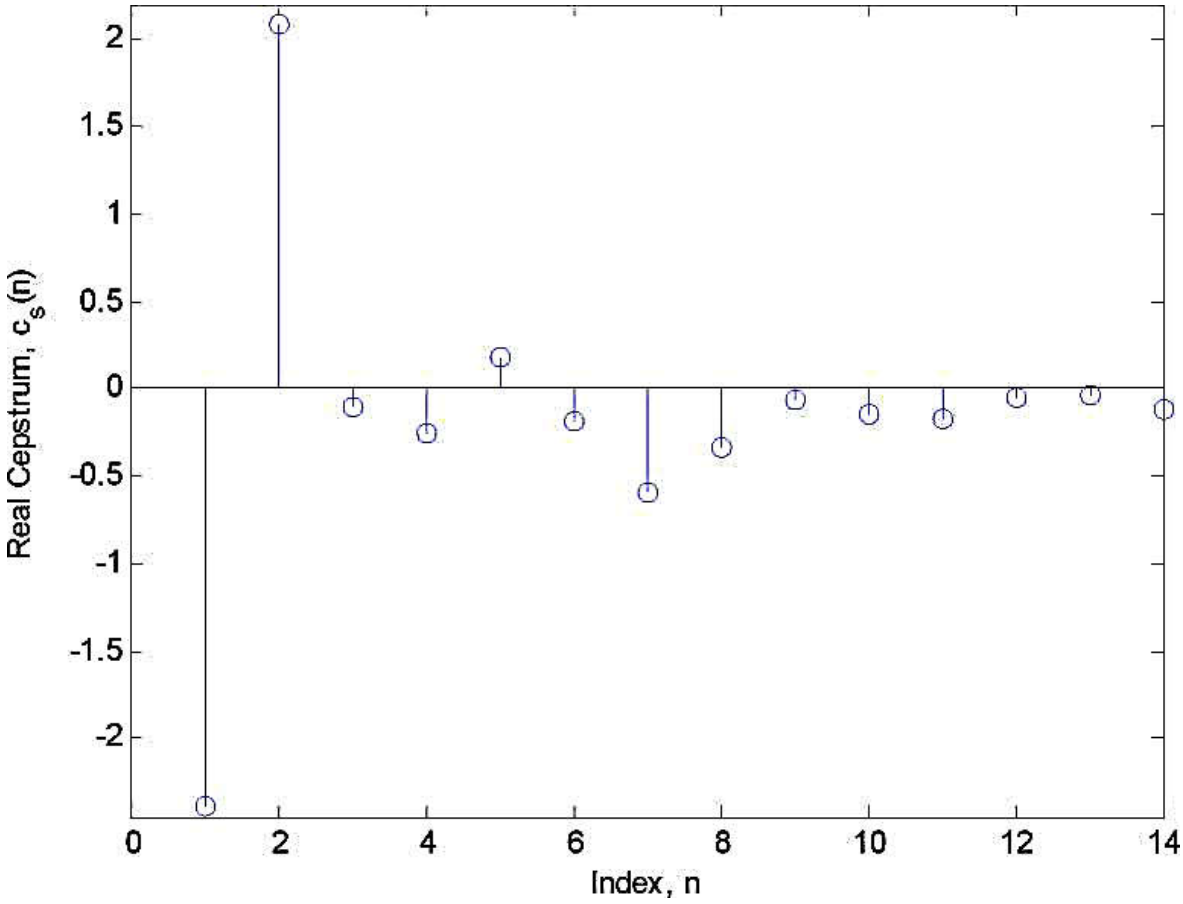


FIGURE 5.3: Real Cepstrum Coefficients of one of the segmented utterances of the word "tem" Only the first 14 coefficients are plotted.

Fig 5.4 shows that the mapping is linear below 1 kHz and is logarithmic above 1 kHz. The mapping is achieved by the formula [Picone, 1993]:

$$Fmel = 2595 \text{ x } \log10\ (1 + f700). \quad (5.7)$$

The MFCCs, however, are computed by using the human perceptual model instead of warping the real cepstrum with a mel scale as discussed above. The human perception, or the operation of the basilar membrane in the inner ear, is generally modelled as a bank of 24 or so bandpass filters. Studies have shown that the basilar membrane is 32 *mm* long and that each bandwidth is about 1.5 *mm* in length along the basilar membrane; resulting in 24 bandpass filters (also called "critical band" filters) needed to model the basilar membrane [Deng, O'Shaughnessy, 2003]. The distribution of these critical band filters is also linear

below 1 *kHz* and logarithmic above 1 *kHz.* Thus, their centre frequencies and bandwidths follow the mel scale shown in Fig 5 4.

The critical band filters are conceptualized by a simple set of triangular windows (or bandpass filters), each cantered on a critical band, as shown in Fig 5 5.

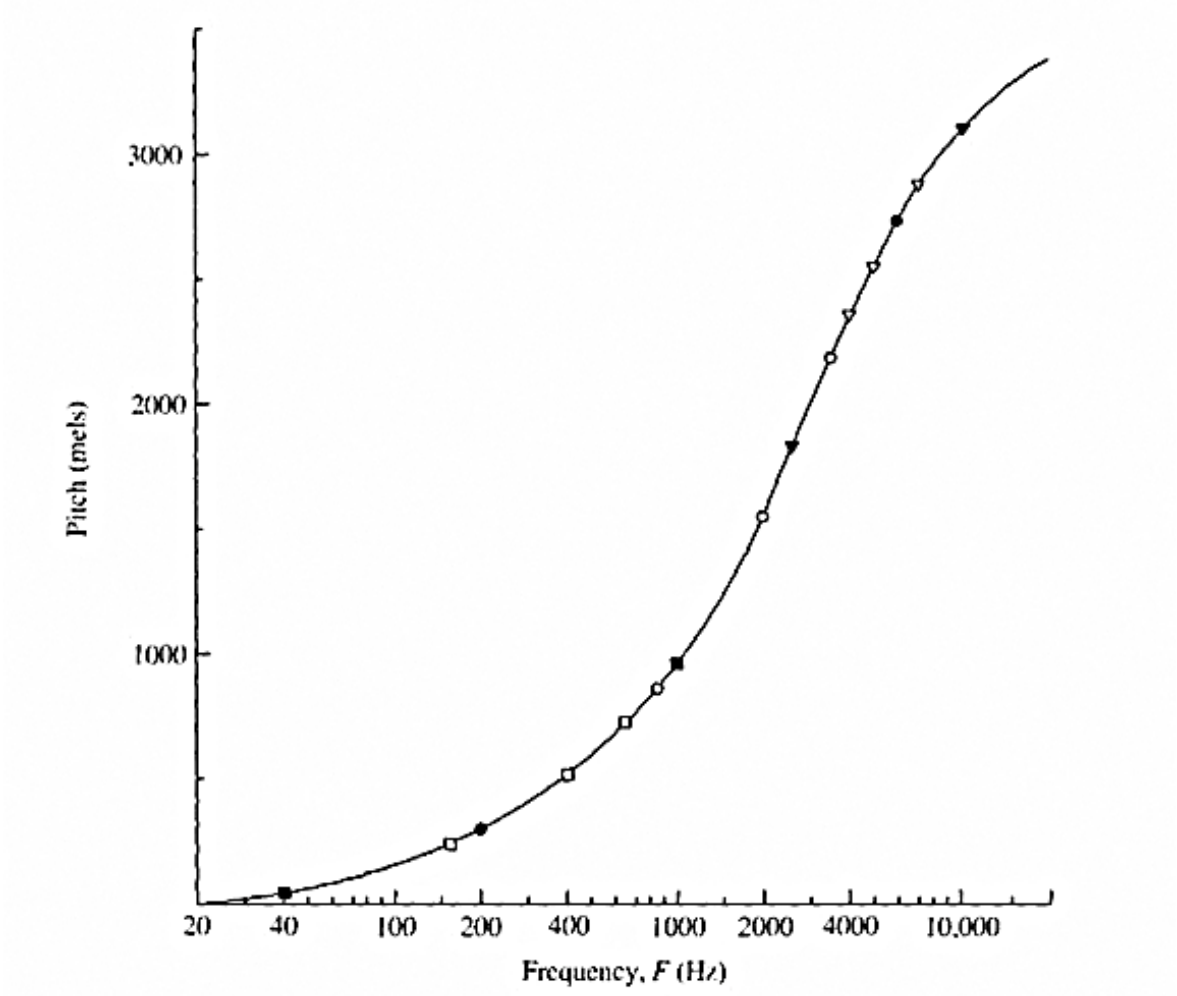


FIGURE 5.4: The mel scale (From: [Deller, Proakis, Hansen, 1993])

In practice, other types of filters can be applied to generate the MFCCs. However, the triangular filters are consistently used in speech recognition studies as they are especially easy to implement [Deller, Proakis, Hansen, 1993]. Thus, triangular filters were chosen to extract the MFCCs in this study.

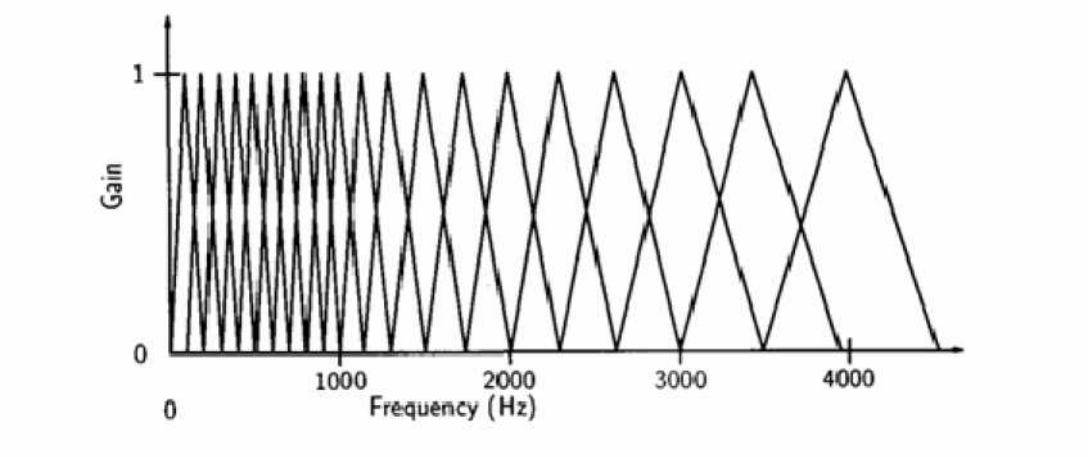


FIGURE 5.5: Conceptual triangular filters for extracting the MFCCs (From: [Deng, O'Shaughnessy, 2003]).

Since the human auditory system responds to the energy in a critical band, the total logarithmic energy in a critical band is obtained through the conceptual filters, and the energy of each critical band is converted too corresponding MFCC via an IDFT. Generally, the final IDFT block is implemented with a discrete cosine transform (DCT) by replacing the complex exponential with a cosine since the logarithmic spectral energy is real and even, as explained in the previous section. The MFCC computational process is presented next and the derivation follows closely that presented in [Vergin, O'Shaughnessy, Farhat, 1999].

First, the spectral magnitude (or energy) of a speech signal or a frame of the speech signal s (n) is calculated as:

$$Si = |S(k)|, \qquad \text{for } i = 0, 1..., N/2 \qquad (5.8)$$

where $S(k)$ is the N -point DFT of the speech signal or a frame of the speech signal:

$$S(K) = \sum_{n=0}^{N-1} log|S(k)|e^{j2\pi kn/N} \qquad \text{for } k = 0, 1..., N-1 \qquad (5.9)$$

The spectral energy, $|S(k)|^2$, can also be used in Equation (8) instead of the spectral magnitude.

Next, the energy in each critical band is obtained by applying the conceptual triangular windows shown in Figure 5 to the spectral magnitude in Equation (5.8):

$$S(k) = \sum_{i=0}^{(\frac{N}{2})-1} S_i h_j(i) \tag{5.10}$$

where J is the total number of triangular filters, $h_J(i)$, used. Finally, MFCCs are calculated as:

$$C_s(n) = \sum_{j=1}^{J} log_{10}(E_j)\cos[n(j+0.5)\frac{\pi}{J}] \tag{5.11}$$

where n is the number of MFCCs to be retained, generally 8 to 14 [Deng, O'Shaughnessy, 2003]. The computational process explained above is illustrated as a lock diagram in Fig 56.

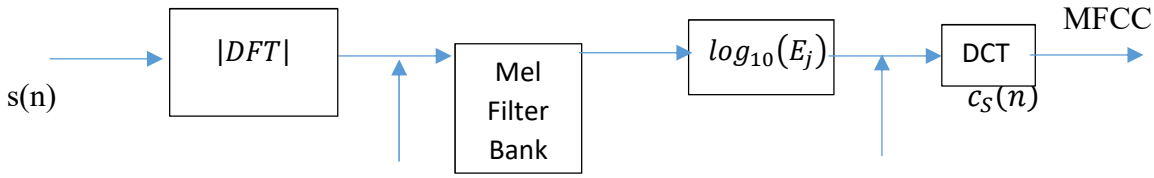


FIGURE 5.6: The MFCC computation as a block diagram (After: [Zhu, Alwan, 2003]).

The first coefficient $c_s(0)$ represents the average power in the speech signal. However, $c_s(0)$ is not often used in recognition applications since the average power varies considerably depending on the microphone placement and channel. The coefficients $c_s(n)$ give increasingly finer spectral details for each $n > 1$ [Deng, O'Shaughnessy, 2003].

Fourteen MFCCs, excluding the first coefficient, are extracted for each segmented utterance and selected as feature vectors for the classification stage of the recognizer. The choice of fourteen MFCCs follows the above discussions, as the coefficients become negligibly small when the order, i.e., index number, increases. Even in the case of fourteen coefficients, the last few coefficients are much smaller in amplitude than the rest of the coefficients.

The MFCCs derived for one of the segmented utterances of the word "up" are illustrated in Fig 5.7.

5.3 Summary

This chapter presented the feature extraction section of the speech recognizer, which serves to represent the segmented utterances in both a useful and efficient way for the recognizer. Specifically, two extensively used spectral features that are also chosen for the current study were presented in detail, namely, the real cepstrum, and the melfrequency cepstrum.

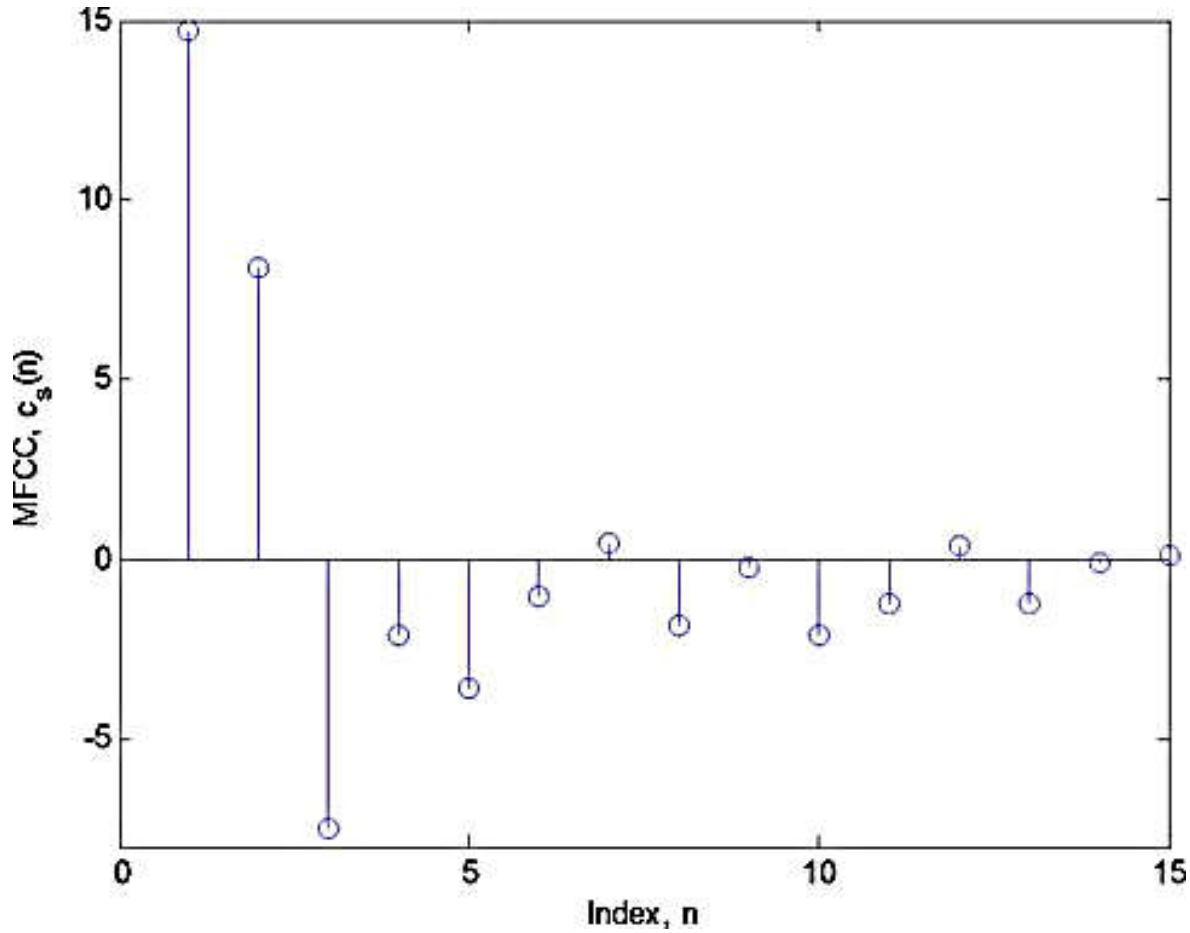


FIGURE 5.7: Mel-frequency Cepstral Coefficients (MFCC) of one of the segmented utterances of the word “upper”

CHAPTER 6

Neural Network Configuration

This chapter describes the feedforward multi-layer neural network configuration used as speech recognizer for this study and the resulting recognition results. First, the chapter introduces the basic concepts behind artificial neural networks. Next, two learning algorithms commonly used in multi-layer neural networks (conjugate gradient and Levenberg-Marquardt algorithms) are presented. Third, we discuss implementation issues for the multi-layer neural networks used for the present study.

6.1 Introduction

Artificial neural networks (ANNs) are inspired by the human nervous system. The human nervous system consists of approximately 1011 nerve cells, or neurons, each of which has 104 connections with other neurons [Deller, Proakis, Hansen, 1992]. A simplified model of a biological neuron is shown in Fig 6 1

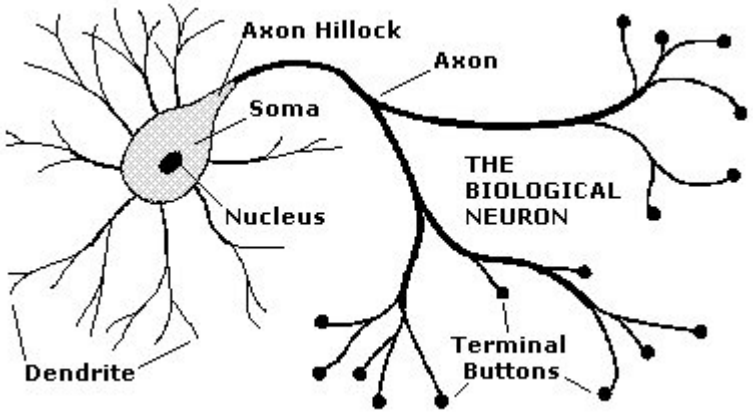


FIGURE 6.1: A simplified model of a biological neuron (From: [Deller, Proakis, Hansen, 1992]).

There are three main components in a nerve cell: the dendrites, the cell body (or the soma), and the axon. The dendrites are the receptive nerve fibres that carry the input signals into the cell body. The cell body sums and thresholds the received signals through the dendrites. The axon is a long transmission line that carries the signals from one cell body to others. The synapse is the connection point between an axon of a cell and a dendrite of another. The nervous system is a highly parallel structure, which is a combination of the nerve cells described above [Hagan, Demuth, Beale, 1996].

ANNs, which are inspired by the biological neural system introduced above, are the simplified version of the complex human nervous system, although the exact mathematical behavior of the nervous system is unknown [Hagan, Demuth, Beale, 1996]. An artificial neuron accepts signals from other neurons or from its inputs, integrates or sums the incoming signals, and then the output is determined according to some sort of threshold function. A typical artificial neuron structure is illustrated in Fig 6.2.

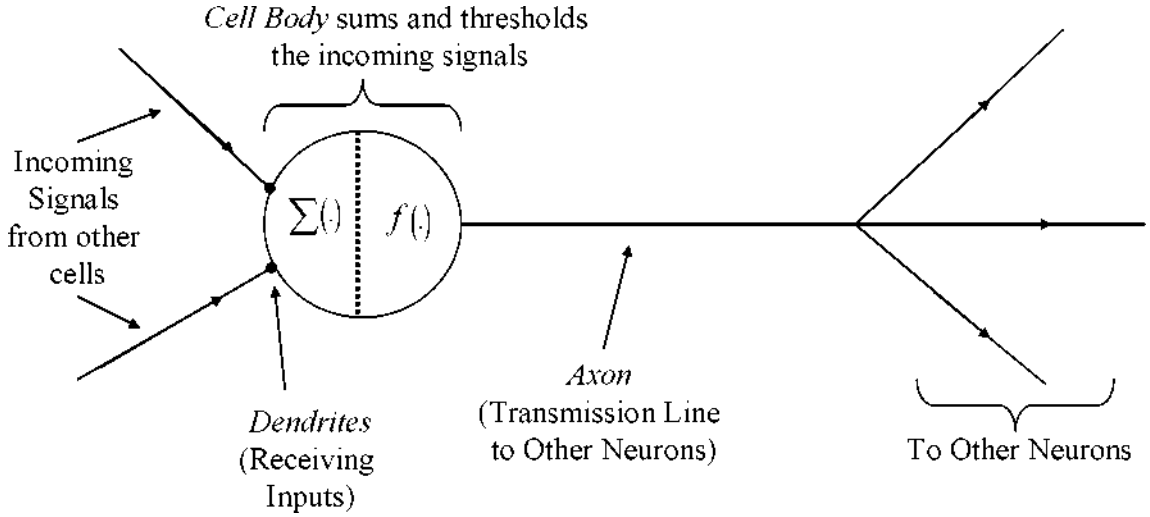


FIGURE 6 2: Artificial neuron model (After: [Deller, Proakis, Hansen, 1992]).

Therefore, an ANN is a network of these artificial neurons connected via some connections in a parallel structure. ANNs learn from examples shown to them, much like the way humans learn from the examples they see, which is called the training phase of the network. After learning from examples during training, they are capable of generalizing the learned examples to new examples that are not introduced during the training period, which is called the testing phase.

The first practical application of artificial neural networks dates back to late 1950s with the invention of the perceptron network by Rosenblatt [Rosenblatt, 1958]. The perceptron was

capable of performing pattern classification on two linearly separable classes. Artificial neural networks did not get much attention until the 1980s mainly due to the lack of powerful computers and processors needed to conduct the experiments combined with the limitations of the networks in classification, as well as the lack of powerful learning algorithms to train the networks for more complex problems. Interest in artificial neural networks increased dramatically with the advances in computing technology. New concepts introduced in the 1980s also contributed to this increase in research efforts on specific ANN types, namely, the recurrent network [Hopfield, 1982], and the backpropagation algorithm [Rumelhart, McClelland, 1986]. In particular, the discovery of the backpropagation algorithm allowed ANNs to perform complex pattern classification tasks on nonlinear data [Hagan, Demuth, Beale, 1996].

Artificial neural networks have found successful applications in a wide variety of fields for the last three decades. These fields include: aerospace, automotive, banking, defense, electronics, entertainment, financial, insurance, manufacturing, medical, oil and gas, robotics, speech, securities, telecommunications, and transportation [Hagan, Demuth, Beale, 1996]. There are many commercial implementations of artificial neural networks that are being used in the related fields mentioned above.

Since the speech recognition can basically be viewed as a pattern classification problem, and since the artificial neural networks are capable of performing complex classification tasks, ANNs have easily become a research tool for speech recognition purposes for the last two decades as an alternative to the hidden Markov model (HMM) that is the most common technique used for speech recognition. Particularly, some advantages of the ANNs which made them attractive for the speech recognition are their flexible architecture, highly parallel and regular structure, robustness to the limited training data, ability to accommodate discriminant learning, and no need to know the statistical distribution of the input features [Morgan, Bourlard, 1995].

ANN applications to speech recognition are basically divided into two broad categories; isolated word recognition and continuous speech recognition. The recognition task in the present study falls into the category of the isolated word recognition.

ANNs have not been quite successful in continuous speech recognition applications, and actually, it is not yet known how to implement a neural network based complete system for continuous speech recognition. This challenge is due to at least one fundamental problem

with the training of the networks used for the continuous speech recognition: a target vector must be defined [Morgan, Bourlard, 1995].

Different types of ANNs, however, have been applied successfully to the phoneme, digit and isolated word recognition, and feedforward neural networks such as multi-layer networks with backpropagation algorithm, radial basis function (RBF) networks, probabilistic neural networks (PNN), and time-delay neural networks (TDNN) are architectures commonly used in these speech applications. A multi-layer neural network configuration with backpropagation algorithm is applied to the isolated word recognition problem of the present study. Next, we briefly introduce the basic idea behind multi-layer neural networks and the backpropagation algorithm.

6.2 Multi-Layer Neural Networks

As indicated earlier, the artificial neuron (also called a node or a unit) is the smallest fundamental building block of an artificial neural network. It is a simple processing unit which tries to mimic the biological neuron.

In an artificial neural network model, incoming signals are multiplied by respective scalar *weights* (or *connections)* and passed to a summer which combines weighted inputs together with a scalar *bias.* The output of the summer forms the *net input,* which can also be viewed as the inner product of the inputs with the weights shifted by some bias (if any). The net input is then applied to an *activation function,* or a *transfer function,* whose output is the output of that specific neuron. The neuron output a is given by:

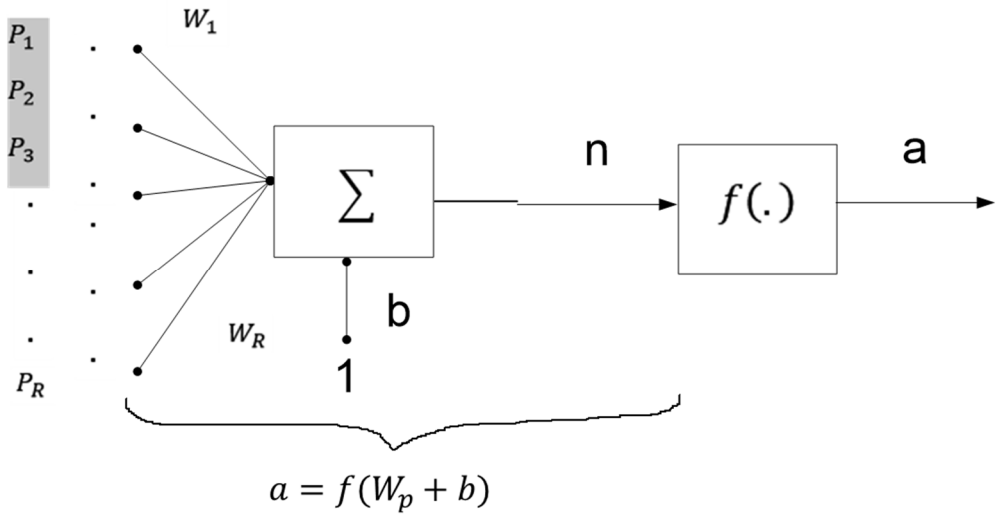
$$a = f(W.p + b) \tag{6.1}$$

where W is the weight matrix, p is the input vector, b is the bias, and $f(.)$ is the activation function.

A single neuron model with multiple inputs is illustrated in Fig 6.3.

When the above model is related to the actual neuron model discussed in the introduction, the weights represent the strengths of the synapses, the summer and the activation function correspond to the cell body, and the net output corresponds to the signals on the axon [Hagan, Demuth, Beale, 1996]. Weights and biases are the values that have to be learned by using a

learning function during training, and need to be stored for use afterwards. The bias, however, may or may not be used.



P_1 to P_R:	input vector (Dendrites)
W_1 to W_R:	weight value…
b:	biased value
Σ and f(.):	activation function (Cell body)
N:	connections (Axons)
A:	outputs

FIGURE 6.3: Mathematical model of a single artificial neuron with multiple inputs (After: [Hagan, Demuth, Beale, 1996]).

The activation function, or the transfer function, can be any type of function that fits the action desired from the respective neuron and is a design choice which depends on the specific problem. Some common types of activation functions are the hard-limit function (*hardlim*), linear function (*purelin*), log sigmoid function (*logsig*), and hyperbolic tangent sigmoid function (*tansig*). Log sigmoid and hyperbolic tangent sigmoid functions are commonly used in multi-layer neural networks with a backpropagation algorithm since they are differentiable and can form arbitrary nonlinear decision surfaces.

The collection of these artificial neurons forms the layers of a network. The neural networks can be classified as single-layer and multi-layer networks depending on the number of the layers. In a multi-layer structure, the last layer is called the output layer, while the rest are called hidden layers since their outputs do not have any connection with the outside environment.

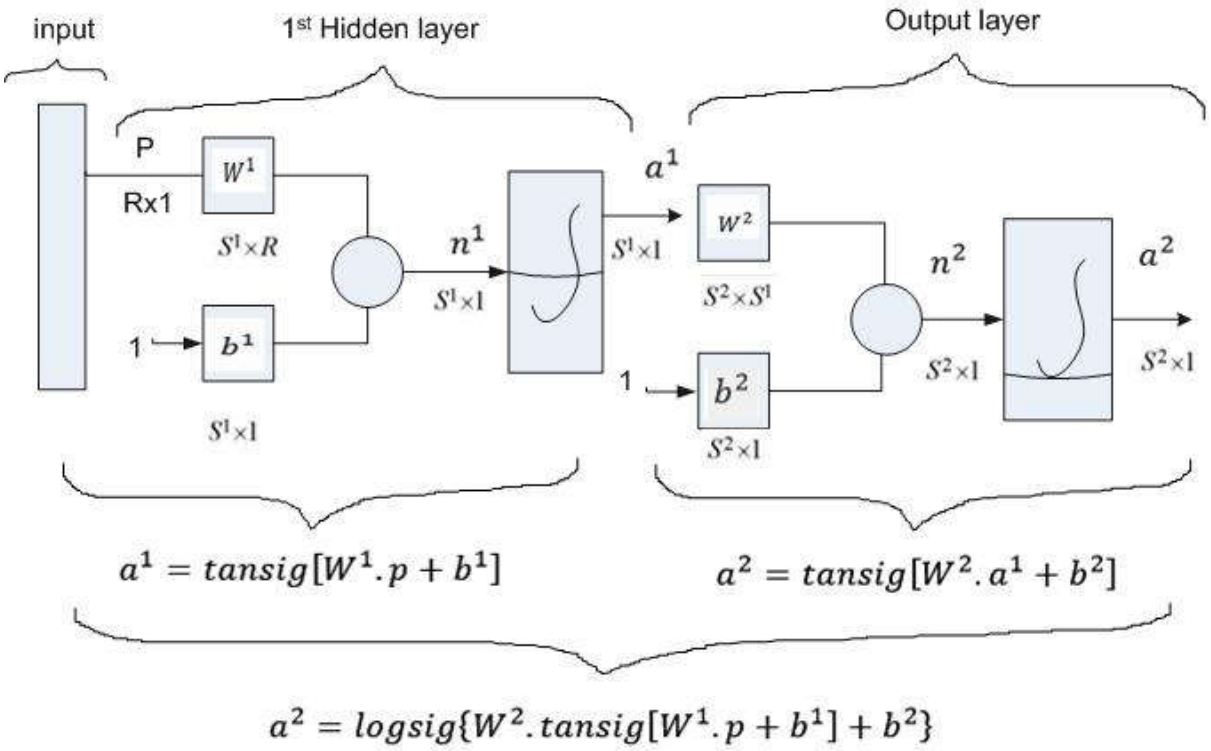


FIGURE 6.4: Feedforward two-layer neural network (After: [Hagan, Demuth, Beale, 1996])

Multi-layer networks are also referred to as feedforward networks since any feedback connection from the output layer to any of the hidden layers does not exist. In a feedforward multi-layer network, the output of a hidden layer effectively becomes an input to the next layer. A feedforward multi-layer neural network that has two layers is shown in Figure 4. The abbreviated matrix notation and network structure in Figure 4 will be used from now on throughout the chapter.

The number of layers, number of neurons in each layer, and type of activation functions per layer or per neuron in each layer are design parameters, i.e., the designer has to choose them according to the specific problem because there are no restrictions or no specific rules about the selection of these parameters. These parameters are generally set on a trial-and-error basis, which may be viewed as a drawback of the neural networks.

Although there is no specific rule on the number of layers to be used, it has been shown that two-layer networks with nonlinearities can, at least in theory, approximate any complex function given a sufficient number of neurons in the hidden layers. This property is the great computational power or expressive power of the multi-layer networks compared to the networks with no hidden layers [Duda, Hart, Stork, 2001].

6.3 The Backpropagation Algorithm

The backpropagation algorithm is the most general and yet powerful method to train a single-layer or a multi-layer neural network. It is a generalization of the least mean square (LMS) algorithm used for linear networks, where the performance index is the mean square error (MSE) for both algorithms.

Basically, a training sequence is passed through the multi-layer network, the error between the target (desired) output and the actual output is computed, and the error is then propagated back through the hidden layers from the output to the input in order to update weights and biases in all layers.

Next, we present the algorithm. Notations and derivations below follow closely [Hagan, Demuth, Beale, 1996], and partially [Duda, Hart, Stork, 2001].

The feedforward operation of a multi-layer network can be defined as:

$$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1}) \qquad \text{for m=0, 1…, M-1} \qquad (6.2)$$

where M is the total number of layers in the network. This is simply the extension of Equation 6.1, to a multi-layer case.

As indicated earlier, the MSE is the performance index (or function) of the algorithm used to update the network parameters. The performance function for a multi-output case is given by:

$$F(x) = E\{e^T.e\} = E\{(t-a)^T.(t-a)\} \qquad (6.3)$$

where t is the target vector, a is the actual output vector, e is the error vector (i.e., the difference between the target and actual outputs), and x is the network parameter vector containing weights and biases.

The expectation operation in Equation (6.3) can be approximated as:

$$\hat{F}(x) = [t(k)-a(k)]^T.[t(k)-a(k)] = e^T(k)e(k) \qquad (6.4)$$

where k is the iteration number.

The weight and bias updates at iteration (k +1) can be expressed in terms of the weights and biases at iteration k, and in terms of the performance function:

$$W_{i,j}^{m}(k+1) = W_{i,j}^{m}(k) - \alpha \frac{\partial \hat{F}}{\partial W_{i,j}^{m}} \tag{6.5}$$

$$b_{i}^{m}(k+1) = b_{i}^{m}(k) - \alpha \frac{\partial \hat{F}}{\partial b_{i}^{m}} \tag{6.6}$$

where $W_{i,j}^{m}$ is the weight associated with the j^{th} connection to the i^{th} neuron at layer m, and α is the *learning rate* that determines the amount of change to the weights and biases. The learning rate also determines how fast the algorithm will converge to the minimum point on the error surface while ensuring convergence.

Note that the last two equations are simply a re-statement of the well-known steepest descent algorithm, where the error is minimized by taking steps of α in the negative direction of the gradient of the performance function which causes a downhill movement on the performance function surface. Thus, the backpropagation algorithm is merely a gradient descent scheme.

The derivations up to this point are identical to the LMS algorithm, which is also a modified version of the steepest descent algorithm. Computing the partial derivatives in Equations (6.5) and (6.6) is the hardest part of the algorithm since the error does not explicitly depend on the weights and biases in the hidden layers. Therefore, the chain rule is used to compute these partial derivatives:

$$\frac{\partial \hat{F}}{\partial W_{i,j}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} X \frac{\partial n_{i}^{m}}{\partial W_{i,j}^{m}} \tag{6.7}$$

$$\frac{\partial \hat{F}}{\partial b_{i}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} X \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \tag{6.8}$$

n_{i}^{m} is the net input of the ith neuron in layer m, and is given by:

$$n_{i}^{m} = \sum_{j=1}^{s^{m-1}} W_{i,j}^{m}.a_{j}^{m-1} + b_{i}^{m} \tag{6.9}$$

Where S^{m-1} represents the total number of neurons in layer m-1.

Using the chain rule makes the error an explicit function of the weights and biases of layer m. Therefore, the partial derivatives of the performance function can be written as:

$$\frac{\partial \hat{F}}{\partial W_{i,j}^{m}} = S_i^m . a_j^{m-1} \tag{6.10}$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = S_i^m \tag{6.11}$$

Where S_i^m is called the sensitivity, which describes how the error changes with the net input at layer m, and is given as [Hagan, Demuth, Beale, 1996]:

$$S_i^m = \frac{\partial \hat{F}}{\partial n_i^m} \tag{6.12}$$

The weight and bias updates defined in Equations (6.5) and (6.6) become in matrix form:

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \tag{6.13}$$

$$b^m(k+1) = b^m(k) - \alpha s^m \tag{6.14}$$

The last two equations above are the weight and bias update equations that define the backpropagation algorithm.

The sensitivities in Equations (6.13) and (6.14) must be computed and propagated back through the network from the output nodes to the input nodes since the error is propagated back with the sensitivities, as will be apparent shortly. This is also obtained using the chain rule:

$$\begin{aligned} S^m &= \frac{\partial \hat{F}}{\partial n^m} \\ &= \left(\frac{\partial n^{m+1}}{\partial n^m}\right)^T \frac{\partial \hat{F}}{\partial n^{m+1}} \\ &= F^m(n^m)(W^{m+1})^T \frac{\partial \hat{F}}{\partial n^{m+1}} \\ &= F^m(n^m)(W^{m+1})^T S^{m+1} \end{aligned} \tag{6.15}$$

where the partial derivative of the net input vector at layer m +1 with respect to the net input vector at layer m is given by:

$$\frac{\partial n^{m+1}}{\partial n^m} = (W^{m+1})F^m(n^m) \tag{6.16}$$

the matrix that contains the partial derivatives of the activations function at layer m is computed as:

$$F^m(n^m) = \begin{bmatrix} f^m(n_1^m) & 0 & \cdots & 0 \\ 0 & f^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f^m(n_{s^m}^m) \end{bmatrix} \tag{6.17}$$

Equation (6.17) reveals an important property of the backpropagation algorithm: all the activation functions used for the network design must be differentiable.

The sensitivity of the output layer which starts the propagation of the network error back to the hidden layers can be computed by using Equation (6.15), and is defined as:

$$S^M = -2SF^M(n^M)\underbrace{(t-a)}_{e} \quad (6.18)$$

The sensitivity at layer m is obtained from the sensitivity of layer m +1; hence, the name backpropagation. The last equation also reveals the fact that the sensitivities are actually a means of propagating the network error back to the hidden layers, and measure how each layer responds to the changes caused by the error. This phenomenon is illustrated in Fig 6.5, where each circle stands for a neuron, and each arrow indicates a connection between neurons of different layers. The weights associated with each connection are also shown in Fig 6.5.

There are a few issues that need to be addressed about the backpropagation algorithm. The very first issue deals with the convergence of the algorithm. There may be cases where the algorithm returns networks parameters which seem to minimize the MSE, but do not yield desired results or approximations. Since the error surface for multi-layer networks is very complex, there are multiple local minimum points in addition to the global minimum along the error surface. In practice, it is impossible to evaluate whether the algorithm converges to the global minimum or a local minimum. Furthermore, it is usually impossible to compute initial weight and bias values close to a minimum. Therefore, it is essential to iterate the

algorithm multiple times over the whole training set with different initial weight and bias values.

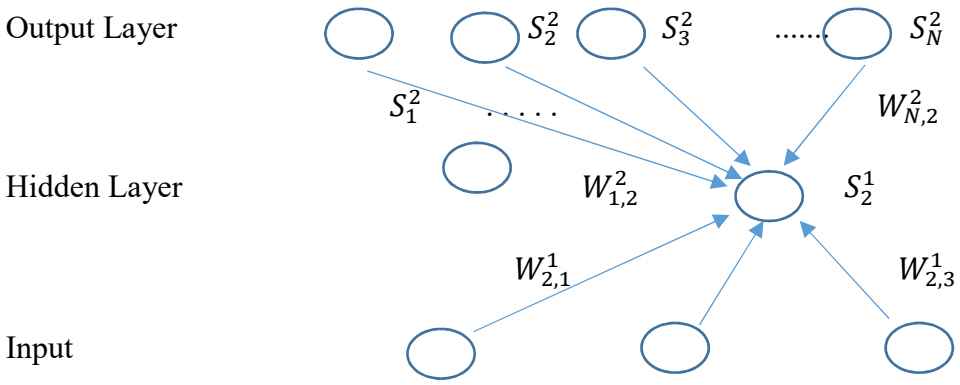


FIGURE 6.5: Backpropagation of sensitivities in a feedforward two-layer neural network (After: [Duda, Hart, Stork, 2001]).

Another issue to consider is the potential long learning time or large number of epochs needed to train the network. Thus, several variations of the basic algorithm have been proposed over the years and successfully used in most of the multi-layer networks with backpropagation to speed-up the algorithm training phase. These modifications and techniques include: adding a momentum term to the algorithm (backpropagation with momentum), variable learning rate, conjugate gradient descent, and Levenberg- Marquardt algorithm.

The conjugate gradient (CG) descent and Levenberg-Marquardt (LM) algorithms will be discussed in the following sections.

6.4 Conjugate Gradient Algorithm

The conjugate gradient (CG) algorithm is a numerical optimization technique designed to speed up the convergence of the backpropagation algorithm. It is in essence a line search technique along any set of conjugate directions, instead of along the negative gradient direction as is done in the steepest descent approach. The power of the CG algorithm comes from the fact that it avoids the calculation of the Hessian matrix or second order derivatives, which are required in the LM derivation, yet it still converges to the exact minimum of a quadratic function with n parameters in at most n steps [Hagan, Demuth, Beale, 1996].

The algorithm starts by selecting the negative gradient direction as the initial descent direction, i.e., the initial search direction. Next, the algorithm moves along the initial search direction until the local minimum in error is reached in that direction. At that point, the next search direction (i.e., the conjugate direction) is computed by selecting a direction orthogonal to the previous one and the following iteration selected as leading to the minimum value along that direction [Duda, Hart, Stork, 2001].

A detailed discussion of the steps taken to accomplish the algorithm will be presented next, where the notations and derivations follow closely those presented in [Hagan, Demuth, Beale, 1996].

The conjugate gradient algorithm starts by selecting the initial search direction as the negative of the gradient:

$$p_o = -g_o \tag{6.19}$$

And

$$g_i = \nabla F(x)|_{x=x_k} \tag{6.20}$$

where x is the vector containing the weights and biases and F(x) is the performance function, i.e., the mean square error (MSE).

The search directions p_i are called conjugate with respect to a positive definite Hessian matrix if and only if

$$p_i^T A p_i = 0 \qquad \text{for } i \neq j \tag{6.21}$$

where A represents the Hessian matrix $\nabla^2 F(x)$.

The above condition can be modified to avoid the calculation of the Hessian matrix for practical purposes, and is given as:

$$\nabla g_i^T p_j = 0 \qquad \text{for } i \neq j \tag{6.22}$$

The new weights and biases are computed by taking a step with respect to the learning rate $\propto_i$ along the search direction that minimizes the error:

$$x_{i+1} = x_i + \propto_i p_i \tag{6.23}$$

where the learning rate $\propto_i$ for the current step is given by:

$$\propto_i = \frac{g_i^T p_i}{p_i^T A_i p_i} \tag{6.24}$$

Next, the new conjugate search direction is selected to continue the algorithm:

$$p_{i+1} = -g_i + \beta_{i+1} p_i \tag{6.25}$$

where the scalar β_i, which can be viewed as a momentum added to the algorithm [Duda, Hart, Beale, 2001], is given by one of three common choices (only Fletcher and Reeves formula is shown here since it is used for the current implementation):

$$\beta_i = \frac{g_{i+1}^T g_i}{g_i^T g_i} \tag{6.26}$$

The algorithm iterates along successive conjugate directions until it converges to the minimum, or a predefined error criterion is achieved.

As is obvious from the above steps, the conjugate gradient algorithm requires a batch mode training, where weight and bias updates are applied after the whole training set is passed through the network, since the gradient is computed as an average over the whole training set [Duda, Hart, Beale, 2001].

The conjugate gradient algorithm outlined above is guaranteed to converge to the minimum in n iterations if the performance function is quadratic with n parameters, as indicated earlier. The algorithm, however, may not converge to the minimum in n iterations if the network is a multi-layer network with many hidden neurons. This is due to the fact that the performance function is not quadratic for multi-layer networks, but may exhibit many local minima. Therefore, the conjugate gradient method was modified to be applied to multi-layer networks. The algorithm does not specify what to do if convergence is not reached after n iterations. As a result, one of the possible approaches to force the algorithm to continue in the case of multi-layer networks is to simply reset the search direction to the negative of the gradient after n iterations [Hagan, Demuth, Beale, 1996].

Although the conjugate gradient algorithm requires many computations to reach convergence, it is one of the fastest batch training algorithms, and has very useful properties, such as avoiding the computation and storage of second order derivatives, while preserving a quadratic convergence property [Hagan, Demuth, Beale, 1996].

6.5 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt (LM) algorithm is a modified version of Newton's method which finds the minimum of a quadratic function in one iteration only by using the second order derivatives information. Newton's method approximates the performance function as a sum of squares, i.e., as a quadratic, which makes the LM algorithm suitable to the training of multi-layer networks as they have complex nonlinear performance surfaces.

The basic Newton's iteration can be written as:

$$x_{i+1} = x_k - \left[\nabla^2 F(x)|_{x=x_i}\right]^{-1} \times [\nabla F(x)] \tag{6.27}$$

where the performance function $F(x)$ is defined as [Hagan, Demuth, Beale, 1996]

$$F(x) = v^T(x).v(x) \tag{6.28}$$

Newton's method requires the computation and storage of the second order derivatives, i.e., the Hessian matrix, as well as the inverse of the Hessian matrix. Newton's scheme is, therefore, expensive and not desirable in large real-life applications. The LM scheme modifies the original scheme given in Equation (6.27) by approximating the Hessian matrix with the Jacobian matrix that contains only first order derivatives.

The weight and bias update formula for the LM scheme can be defined as [Hagan, Demuth, Beale, 1996]:

$$x_{i+1} = x_k - [J^T(x_i)J(x_i) + \mu_i I]^{-1} J^T(x_i)v(x_i) \tag{6.29}$$

Where μ_i is small a small scalar added to the Hessian matrix estimation in order to insure it is invertible, and J (x) represents the Jacobian matrix containing the first order derivatives [Hagan, Demuth, Beale, 1996]:

$$J(x) = \begin{bmatrix} \frac{\partial v_1(x)}{\partial x_1} & \cdots & \frac{\partial v_1(x)}{\partial x_N} \\ \vdots & \vdots & \vdots \\ \frac{\partial v_N(x)}{\partial x_1} & \cdots & \frac{\partial v_N(x)}{\partial x_N} \end{bmatrix} \tag{6.30}$$

The algorithm approaches the steepest descent algorithm with a small learning rate when μ_i increases, whereas it approaches Newton's scheme when μ_i decreases. The LM algorithm, therefore, combines the speed of Newton's scheme and the guaranteed convergence of the steepest descent [Hagan, Demuth, Beale, 1996].

The LM scheme is the fastest algorithm among the possible selection of algorithms for networks with moderate to small size parameters. However, the algorithm has two main drawbacks. First, the algorithm is computationally intensive as it requires more computations per iteration, including the matrix inversion, than other schemes such as the conjugate gradient. Second, it requires the storage of the Hessian matrix estimation, which is a n x n matrix, where n represents the number of network parameters (i.e., the weights and biases), whereas other schemes such as the conjugate gradient requires only the storage of the gradient that is a n dimensional vector. As a result, it becomes impractical to use the LM scheme for large network configurations [Hagan, Demuth, Beale, 1996].

6.6 Implementation

Multi-layer neural networks with the backpropagation algorithm are used in this speech recognition study. Two different neural network structures were implemented: a two-layer feedforward neural network with one hidden and one output layer, and a three- layer feedforward neural network with two hidden layers and one output layer. Note that the two neural network structures considered in this study use same inputs, target assignments, activation functions, output layer structure, network parameters, and differ only in the number of layers and hidden neurons in the hidden layers.

For both network types, the output layer has seven neurons, each of which corresponds to one word in the vocabulary. Recall from Chapter II that the vocabulary chosen for this study has only seven words. For the classification purpose, each word is assumed to correspond to

a class, and each word belonging to its respective class is labelled with an integer number from one to seven. More on the class and target assignments will be explained later.

The numbers of hidden neurons in the hidden layers were varied in each trial of two multi-layer networks in order to evaluate their performance. For the two-layer network, 50, 100, and 150 hidden neurons in the hidden layer were selected in the implementation. Therefore, the two-layer networks employed for this study are denoted as (50-10), (100-10), and (150-10). Four different structures with different number of hidden neurons were implemented for the three-layer network structure: (30 - 20 - 10), (40-20-10), (50-30-10), and (60-40-10).

The hyperbolic tangent sigmoid function (*tansig*) was selected as the activation function for the hidden neurons, as it provides the necessary nonlinearities in the network to solve the classification problem. The log sigmoid function (*logsig*) was used for the neurons at the output layer in order to restrict the network outputs to the interval [0,1].

Recall that both real cepstrum (RC) coefficients and mel-frequency cepstral coefficients (MFCCs) were extracted from each segmented utterance to be used as feature vectors. The MFCCs are the primary features used as the input vectors that represent the segmented utterances for all the multi-layer networks implemented. In addition, the RC coefficients are used with one of each of the two-layer and three-layer configurations, namely the (150 - 10) and (60 - 40 - 10) networks for comparison purposes. As explained in Chapter IV, each segmented speech is represented with 14 spectral coefficients, either MFCC or RC coefficients. Therefore, the network input vectors which represent the segmented utterances are 14 dimensional, i.e., 14 x1 column vectors. Finally, inputs were pre-processed to have zero mean and unit variance before being fed into the network to enhance the network performance.

The architectures of the multi-layer neural networks implemented for the word recognition are shown in Fig 6.6 and Fig 6.7, which illustrate the (150-10) and (60¬40-10) configurations.

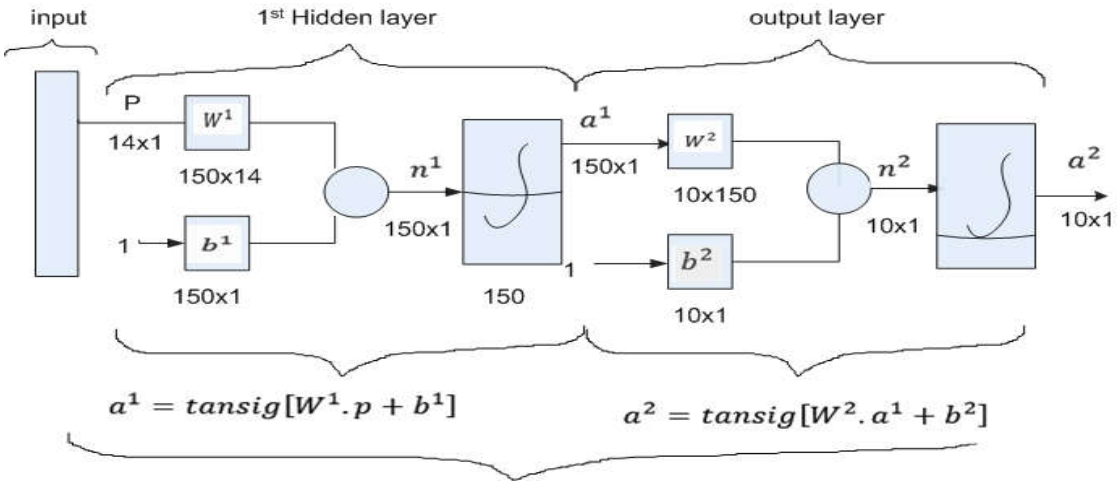


FIGURE 6.6: Two-layer feedforward neural network architecture implemented; (150 - 10) configuration

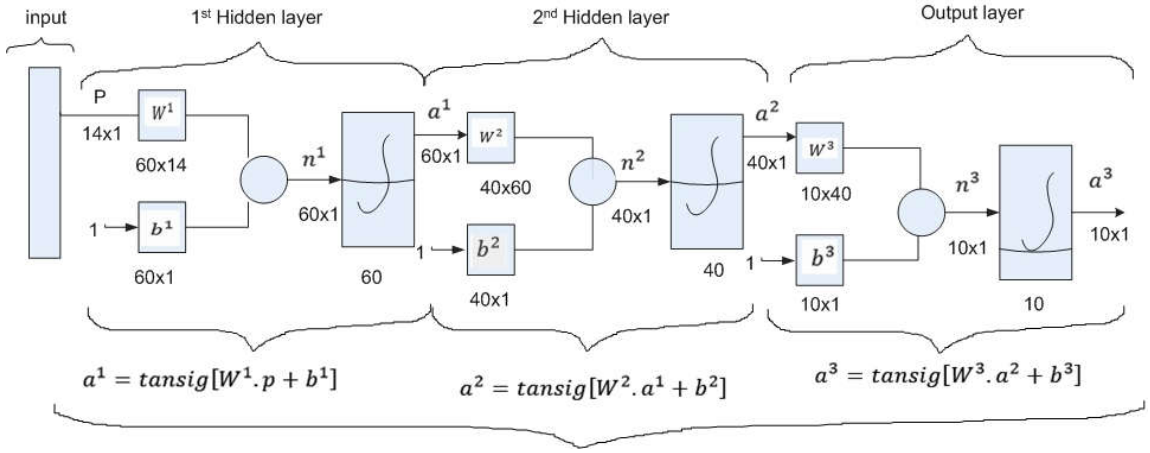


FIGURE 6.7: Three-layer feedforward neural network architecture implemented; (60-40 - 10) configuration

1-of- n coding was selected for the target representation, where n = 10 is the total number of classes. In the 1-of-10 representation, the output of one of the neurons at the output layer which corresponds to one of the seven classes is set to one, with the output of the rest set to zero. For instance, the word "ડાબી" is labelled as Class 1, and its associated target vector is defined as $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$. The class number and target vector assignment for each word in the vocabulary are shown in Table 6.1

Network outputs are continuous between zero and one, as the *logsig* function is used in the output layer. In order to achieve 1-of- n coding, network outputs were converted to zeros and ones by passing them through a simple maximum detector which assigns one to the maximum output value and zero to the rest.

TABLE 6.1: Class numbers and target vectors associated with the vocabulary words

Vocabulary Words	Class Number	Associated Target Vector
ડાબી	1	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]^T$
જમણી	2	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0]^T$
ઉપર	3	$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$
નીચે	4	$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$
આગળ	5	$[0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0]^T$
પાછળ	6	$[0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]^T$
આજુ	7	$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0]^T$
બાજુ	8	$[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
આમ	9	$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$
તેમ	10	$[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

The conjugate gradient (CG) algorithm was selected as the main backpropagation learning function instead of the Levenberg-Marquardt (LM) algorithm because the CG approach was computationally much faster and led to better classification results. Recall that the CG algorithm possesses two useful properties: the memory requirement is minimal when compared to the LM algorithm, and the CG method is much faster than the LM scheme although it usually requires a larger number of epochs for convergence. However, we experimentally observed that the longer the network was trained, the better the results were. One of the multi-layer network configurations, namely the (40 - 20 - 10) network, was also tested with the LM scheme to compare the results of the LM scheme with those obtained with the CG scheme. All these issues will be addressed in the next section.

All network configurations considered in the study were applied to 80 experiments (i.e., iterations) to obtain statistically meaningful results. As will be explained later in this section, the training phase used a randomly selected fixed percentage of the data, while the rest was used for the testing phase. The maximum number of epochs for network training was set to 1,000 after observing that convergence was reached within that range.

One of the problems associated with the neural network training is called overfitting, where the error on the training set approaches to zero, but the error becomes very large on the testing set as the network memorizes patterns shown in the training set but is unable to generalize to slightly different patterns contained in the testing set and to classify new samples in the testing set. As a result, the best way to tackle the generalization issue is to select a network configuration that is just large enough to provide the desired result.

However, determining the right size of a multi-layer network beforehand is very difficult unless the problem is easy to solve, and the network size is usually determined experimentally. Two commonly used methods to improve generalization are regularization and early stopping. In this study, we selected the regularization technique to prevent overfitting. Regularization improves generalization by modifying the performance function (i.e., the MSE) and adding an additional term that contains the mean of the sums of the network parameters.

The modified performance function msereg is given as [Demuth, Beale, 2005]:

$$msereg = \gamma mse + (1 - \gamma) msw \tag{6.31}$$

where γ is the performance ratio, which is also determined experimentally, and set to 0.85 for this study. Using *msereg* causes the weights and biases to be smaller, which in turn yields a smoother network response that is better designed to avoid overfitting [Demuth, Beale, 2005].

TABLE 6.2: Multilayer structures studied.

Structure	Neuron #s	Features	Activation Function	Training Function	Perform. Function	Iteration #s
Two-layer	50-10	MFCC	tansig - logsig	CG	msaereg	80
	100-10	MFCC		CG		
	150-10	MFCC & RC		CG		
Three-layer	30-20-10	MFCC	tansig - tansig - logsig	CG		
	40-20-10	MFCC		CG & LM		
	50-30-10	MFCC		CG		
	60-40-10	MFCC & RC		CG		

The training set selected for each iteration was formed by randomly picking 15 repetitions of a word for each subject. As a result, the total size of the training set was 2100, or (20x 10x15), since there are 20 subjects and 10 words in the vocabulary. The remaining repetitions of a word for each subject were assigned to the testing set for each experiment (i.e., iteration). As a result, the relative percentages of the training and testing sets were 25.52% and 74.48%, respectively. The total data size generated using different combinations of word and persons is 8,228. So, it resulting in the total size of 6,128 for the testing set.

The various network configurations considered in this study are listed in Table 6.2.

6.7 Summary

- This chapter covers basic ideas of the feed forward multi-layer neural network configuration used for the speech recognition system.
- The basic concepts behind artificial neural is explained.
- Four different types of the activation functions are tested: Hard-limit function (*hardlim*), Linear function (*purelin*), log sigmoid function (*logsig*) and hyperbolic tangent sigmoid function (*tansig*).
- Log sigmoid and Hyperbolic tangent sigmoid functions commonly used in multi-layer neural network with back propagation algorithm since they are differentiable and can form arbitrary nonlinear decision surface.
- Hyperbolic tangent sigmoid function (*itansig*) was selected as the activation function for hidden neuron as it provides necessary nonlinearities in the network to solve the classification problem.
- Log sigmoid function (*logsig*) was used at output layer in order to restrict the network output to interval [0,1].
- Two learning algorithms namely, Conjugate Gradient (CG) and Levenberg-Marquardt (LM) are explained in detail.
- The conjugate gradient (CG) algorithm was selected as the main backpropagation learning function instead of the Levenberg-Marquardt (LM) algorithm because the CG approach was computationally much faster and led to better classification results.
- Experimentally its observed that longer the network trained batter the results are obtained.
- The basic problem of the neural network trainings is *overfitting*, where error on the training set approaches to zero, but the error becomes very large on the testing set as the network memorizes patterns shown in the training set but is unable to generalize to slightly different patterns contained in the testing set and to classify new samples in the testing set.
- Two commonly used methods to improve generalization are regularization and early stopping.

- In this study, the regularization technique is used to prevent overfitting.
- Regularization improves generalization by modifying the performance function (i.e., the MSE) and adding an additional term that contains the mean of the sums of the network parameters.

CHAPTER 7

Recognition Results

The word recognition results obtained with the different multi-layer neural network configurations considered are presented in this section. Performance measures include:

- The confusion matrix for the training set,
- The confusion matrix for the testing set,
- The confusion matrix when the network is tested on "ડાબી બહાર" and "જમણી બહાર" on which it is not trained,
- Average classification rate and 95% confidence interval plot for the testing set,
- Average classification rate and 95% confidence interval table for the testing sets of all the configurations used in the study.

7.1 Network Configurations Considered

Table 7.1 shows the overall average classification rates for both the training and testing sets, and the 95% confidence intervals for the average classification rates for the testing sets obtained after 80 iterations with each network configuration considered in the study. Results show that overall average classification rates increase for two-layer and three-layer network configurations as the number of neurons in the hidden layers increases. The best two-layer network average classification rates are obtained with the (150 - 10) configuration, and obtained with the (60 - 40 - 10) configuration among the three-layer networks. Results also show their performances to be very similar with 94.731% and 94.61%, for the two- and three-layer network structures, respectively. Results in Table 1 also show that MFCCs lead to better recognition rates than RCs do. We note there is about an 8% difference between the average classification rates obtained with the best network structures, i.e., the (150 - 10) and

(60 - 40 - 10) networks, using the MFCCs and RCs as features. Another important point to note is the performance difference between the network trained using the CG or the LM scheme. The network trained using the CG algorithm yields around a 3.5% higher recognition result than that obtained with the LM algorithm on the same network configuration, i.e., the (40 - 20 - 10) network that operates on the MFCC. Results also show that the 95% confidence interval (CI) for the LM scheme is the largest of all CIs obtained with MFCCs as input features in this study, which makes the network configuration obtained with the LM scheme much less desirable.

TABLE 7.1: Average recognition results obtained for the different multi-layer neural network configurations considered in this study.

Network configuration	Activation function	Training function	Features	average classification rate for training set	average classification rate for testing set
50-10			MFCC	84.72	80.76
100-10	Tansig-		MFCC	87.44	82.22
150-10	logsig	CG	MFCC	88.58	84.73
150-10			RC	85.4	76.29
30-20-10			MFCC	87.68	81.6
40-20-10			MFCC	87.42	83.68
50-30-10	Tansig-Tansig-logsig	CG	MFCC	87.01	84.7
60-40-10			MFCC	89.58	84.92
60-40-10			RC	82.56	76.52
40-20-10		LM	MFCC	87.02	78.74

Table 7.1. Average recognition results obtained for the different multi-layer neural network configurations considered in this study.

7.2 Computational Time Issues

One last note that has to be addressed about the use of LM algorithm is the choice of the (40-20-10) network with the LM scheme and the amount of time needed to complete all 80 experiments with the LM scheme. The choice of the (40 - 20 - 10) network is due to the large memory requirements of the implementations with the LM scheme, as discussed in earlier sections. As a result, that network configuration was the largest complexity network we could run with LM scheme with an Intel core I5 processor with 2GB RAM for 80 iterations without early termination due to “out-of-memory” problems. We noted that using the reduced memory LM option in the LM algorithm did not help with the out-of-memory

problem for the configurations considered. We also noted that the CG algorithm applied to the same complexity network considered in this study, (40 - 20 - 10), converged with about 1000 epochs in about eight minutes, while the LM algorithm took on average 15 minutes to compute 20-30 epochs. Therefore, the time required for a multi-layer network with LM algorithm to complete an implementation with multiple iterations was not very practical as the network complexity increased.

7.3 Results

Recognition results obtained with each multi-layer network configuration shown in Table 1 will be presented in this section. Confusion matrices for both training and testing sets, confusion matrices for "Dabi our" and "Jamni out" are given in Tables 7.2 through Table 7.31. Average classification rate and 95% confidence intervals for the testing sets of all configurations used in the study are presented in Table 7.31.

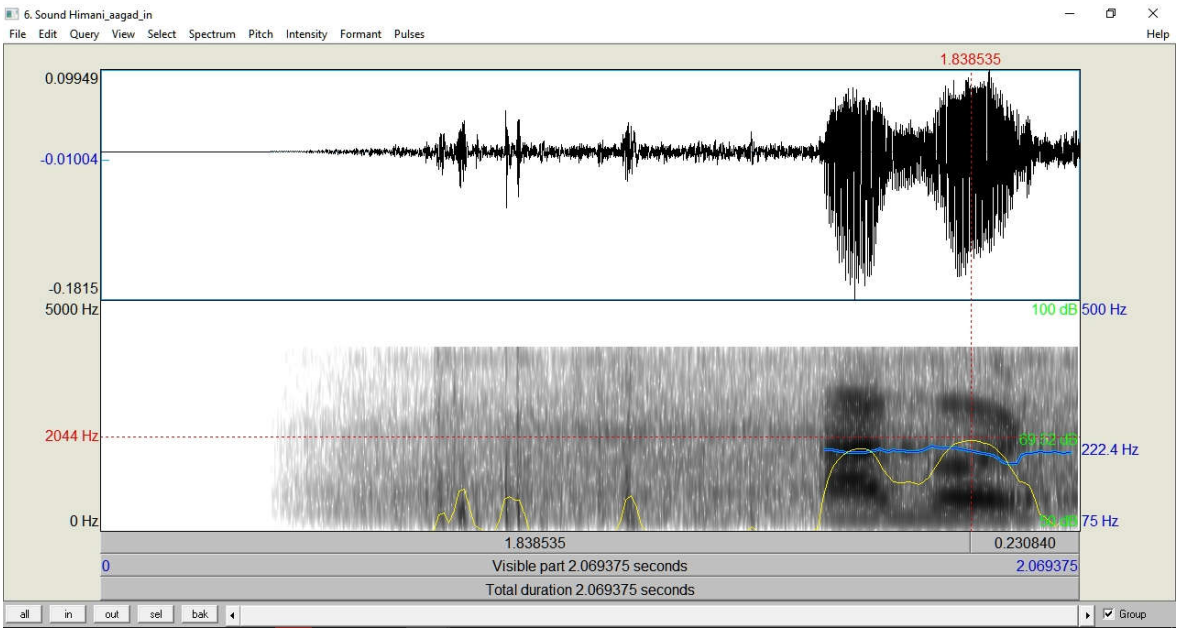


FIGURE 7.1: Waveform and spectrogram for the word "agad" by keeping in-ear microphone.

Short Time Fourier Transform (STFT) is used, to extract spectral characteristics of the speech dataset. It will be represented by the spectrogram, as speech is combinations of frequency dependent parameters. The recorded utterance is sampled at 8 kHz rate so; frequency axis of the spectrogram will go up to 4 kHz. The time axis will go up to the length of the recording signal.

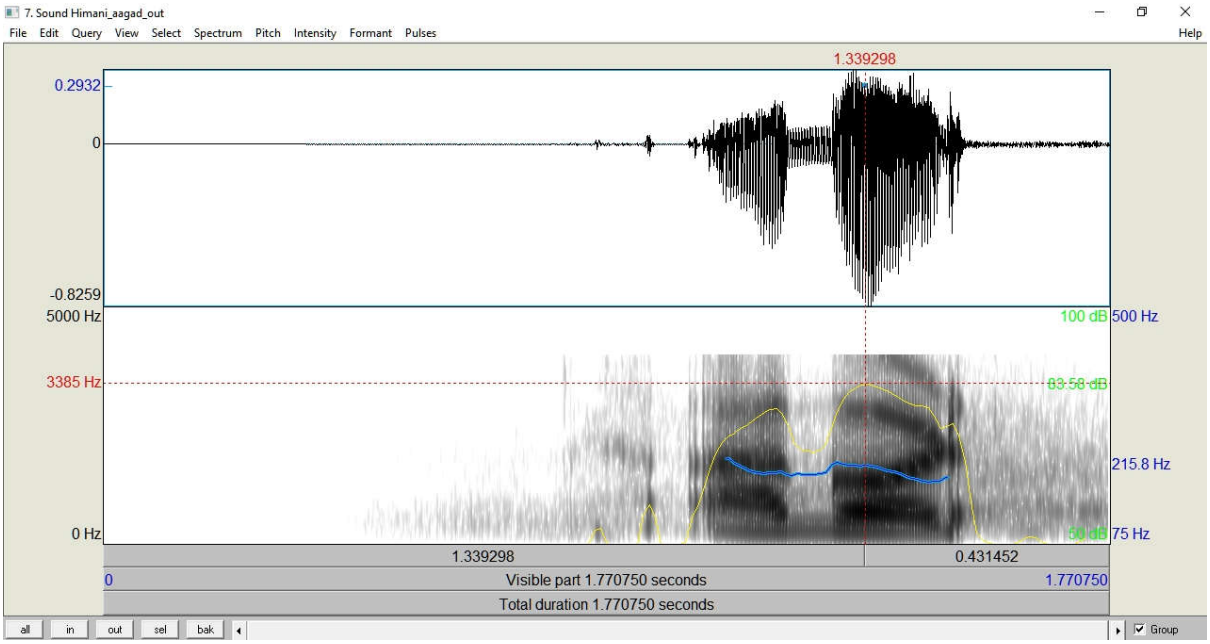


FIGURE 7.2: Waveform and spectrogram for the word "agad" by keeping the microphone outside the mouth.

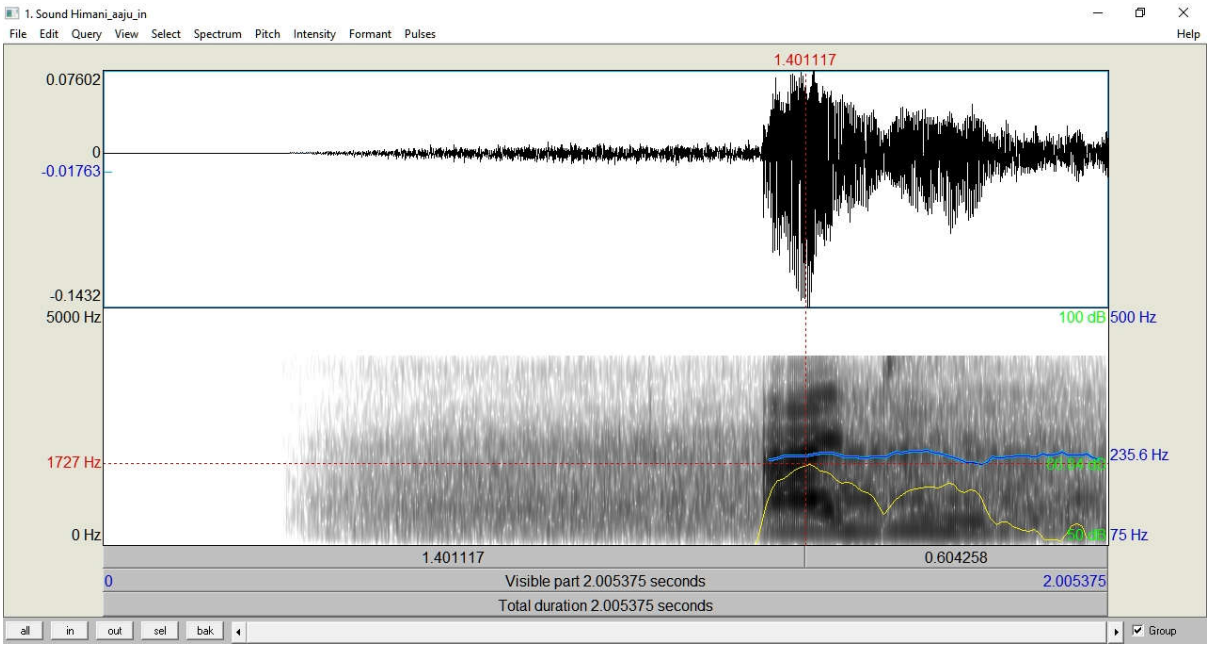


FIGURE 7.3: Waveform and spectrogram for the word "aaju" by keeping in-ear microphone.

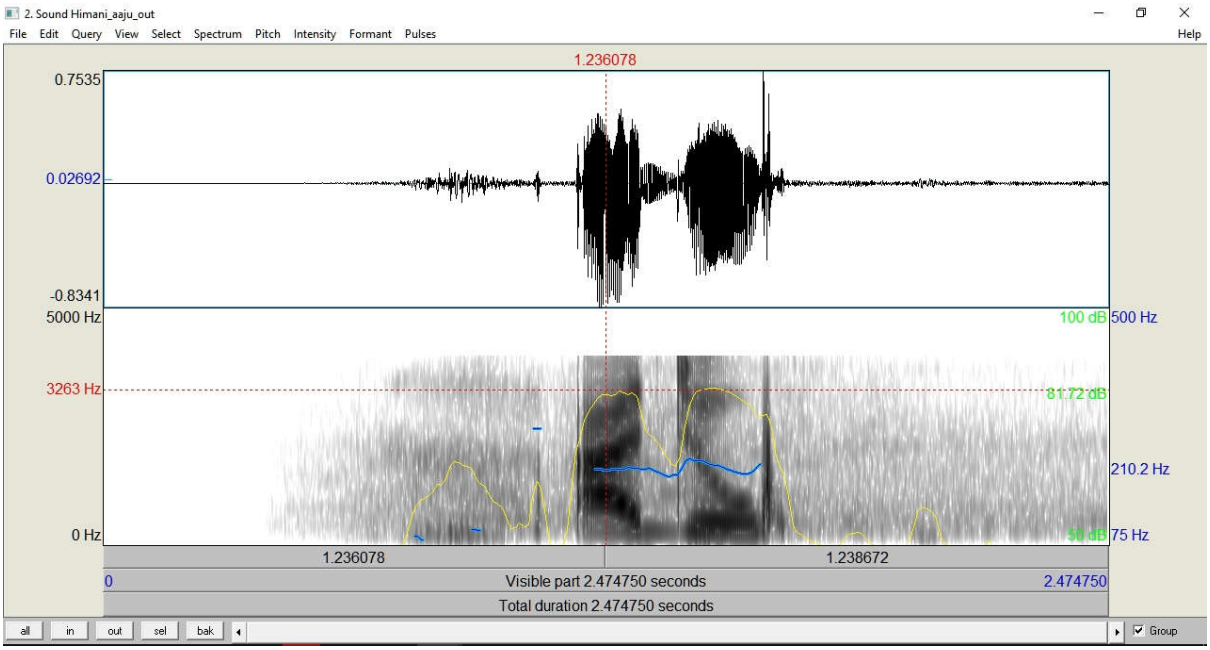


FIGURE 7.4: Waveform and spectrogram for the word "aaju" by keeping the microphone outside the mouth.

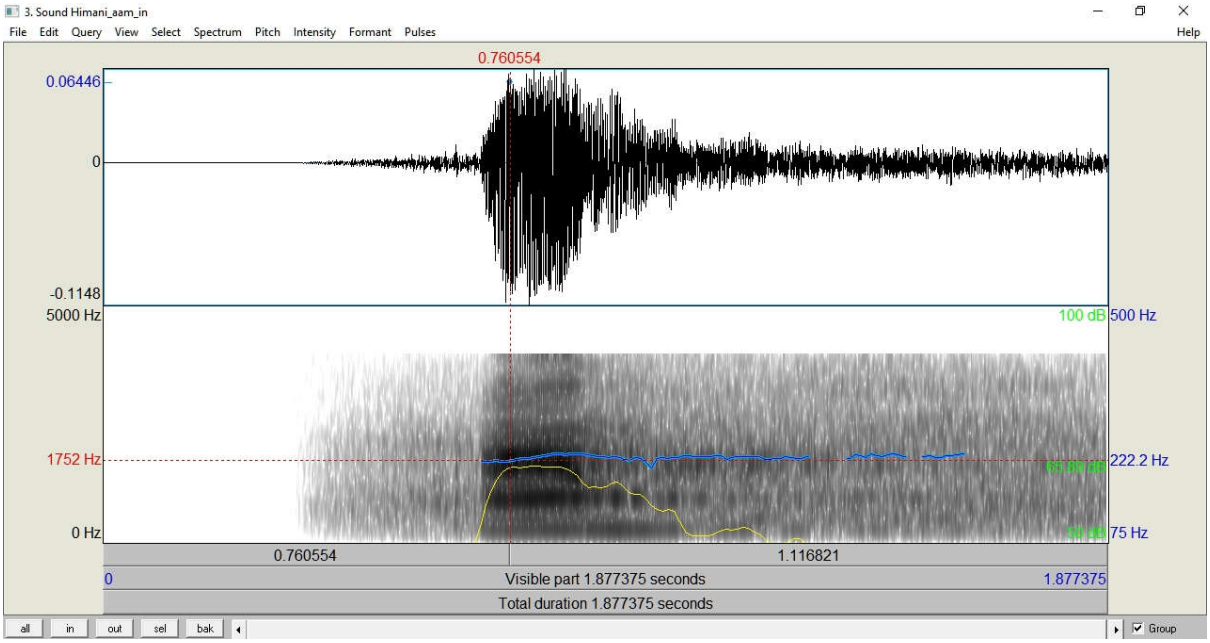


FIGURE 7.5: Waveform and spectrogram for the word "aam" by keeping in-ear microphone.

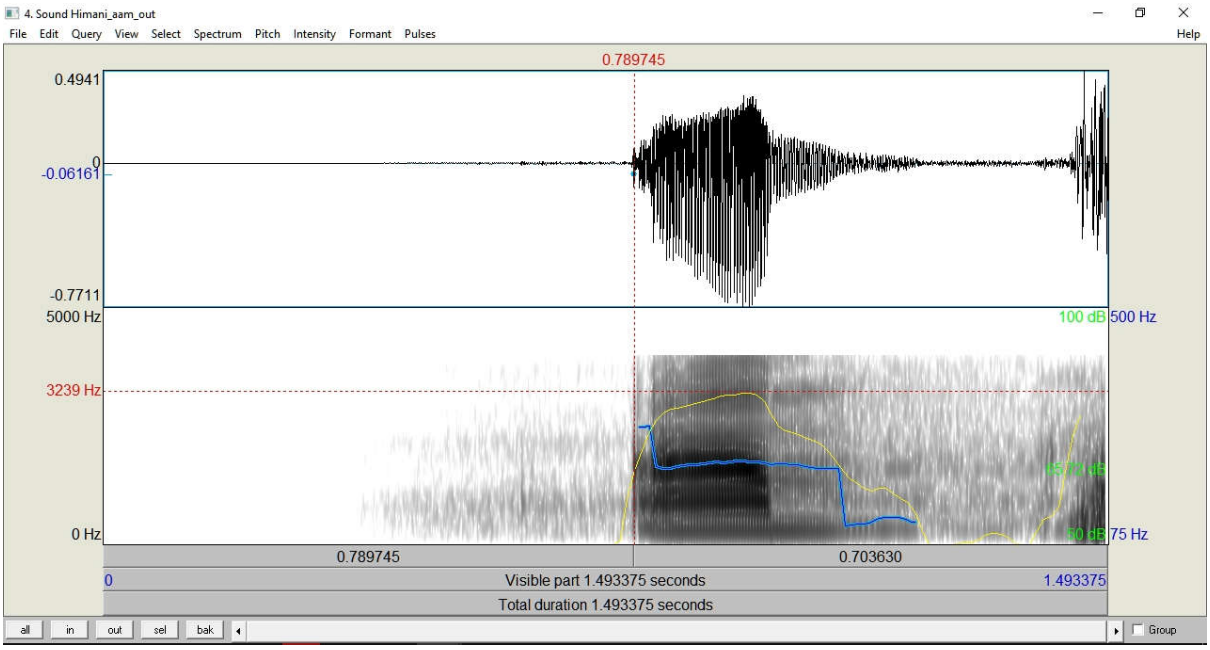


FIGURE 7.6: Waveform and spectrogram for the word "aam" by keeping the microphone outside the mouth.

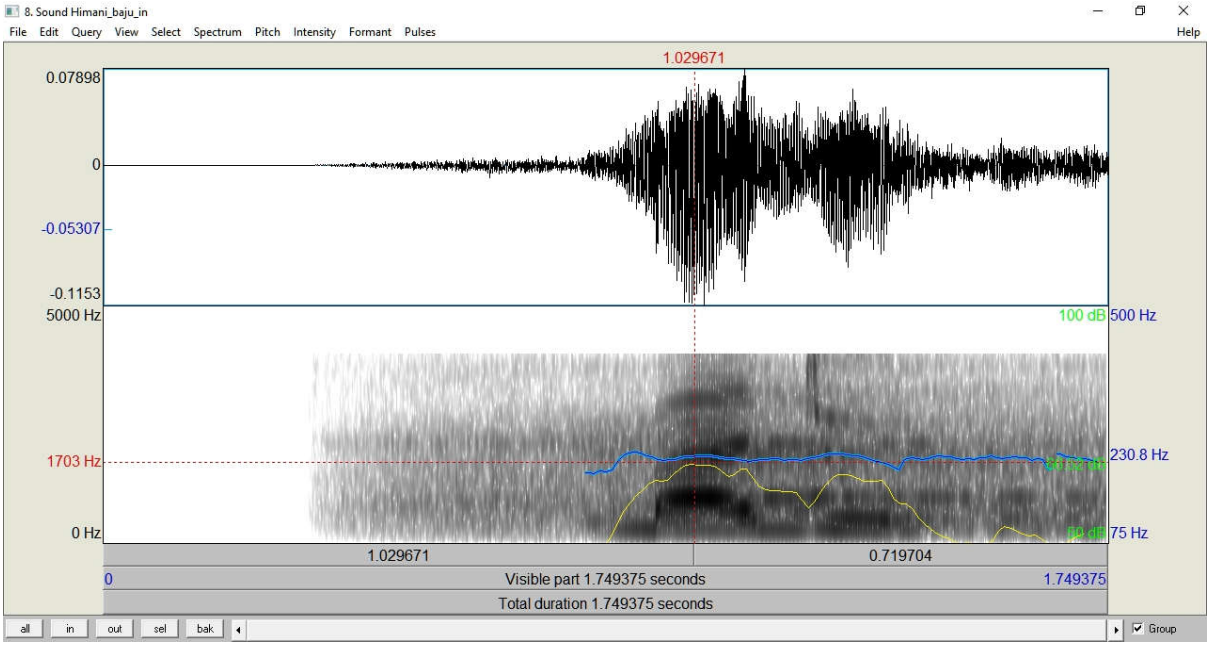


FIGURE 7.7: Waveform and spectrogram for the word "baju" by keeping in-ear microphone.

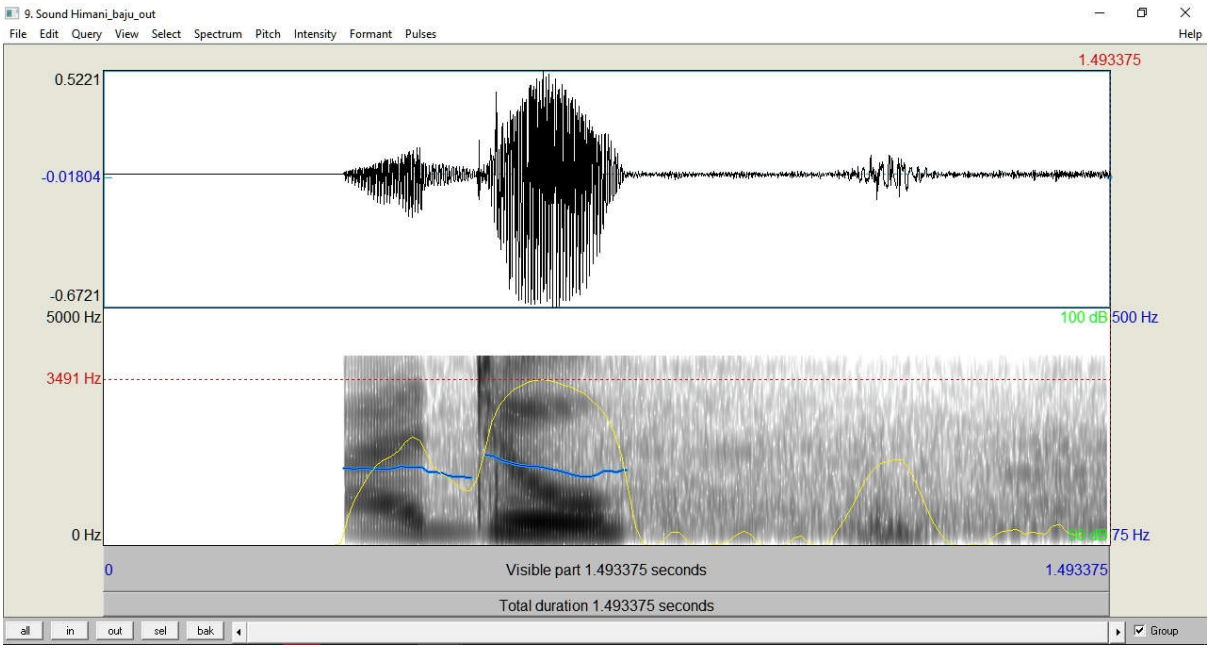


FIGURE 7.8: Waveform and spectrogram for the word "baju" by keeping the microphone outside the mouth.

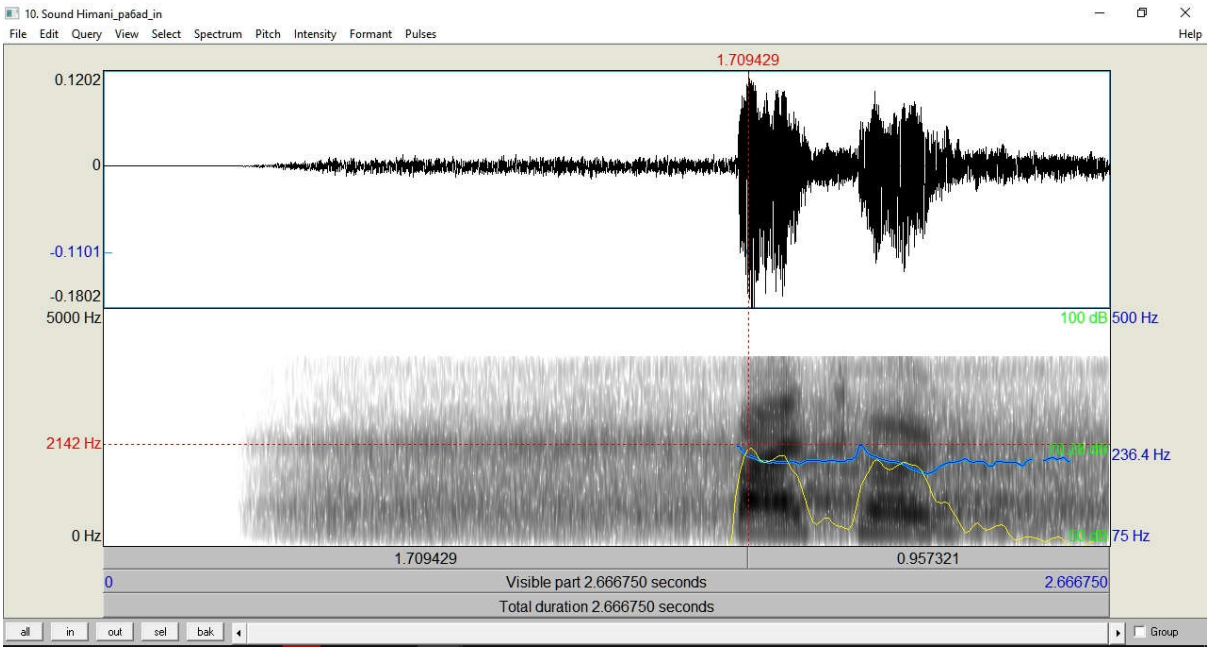


FIGURE 7.9: Waveform and spectrogram for the word "pachhad" by keeping in-ear microphone.

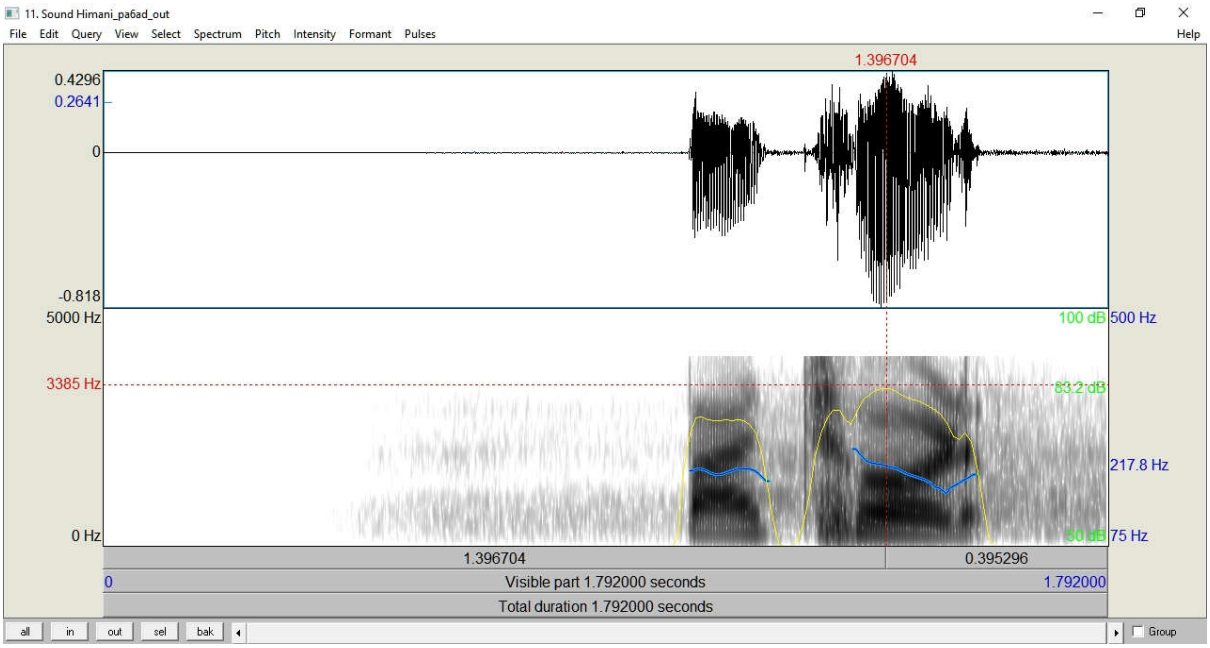


FIGURE 7.10: Waveform and spectrogram for the word "pachhad" by keeping the microphone outside the mouth.

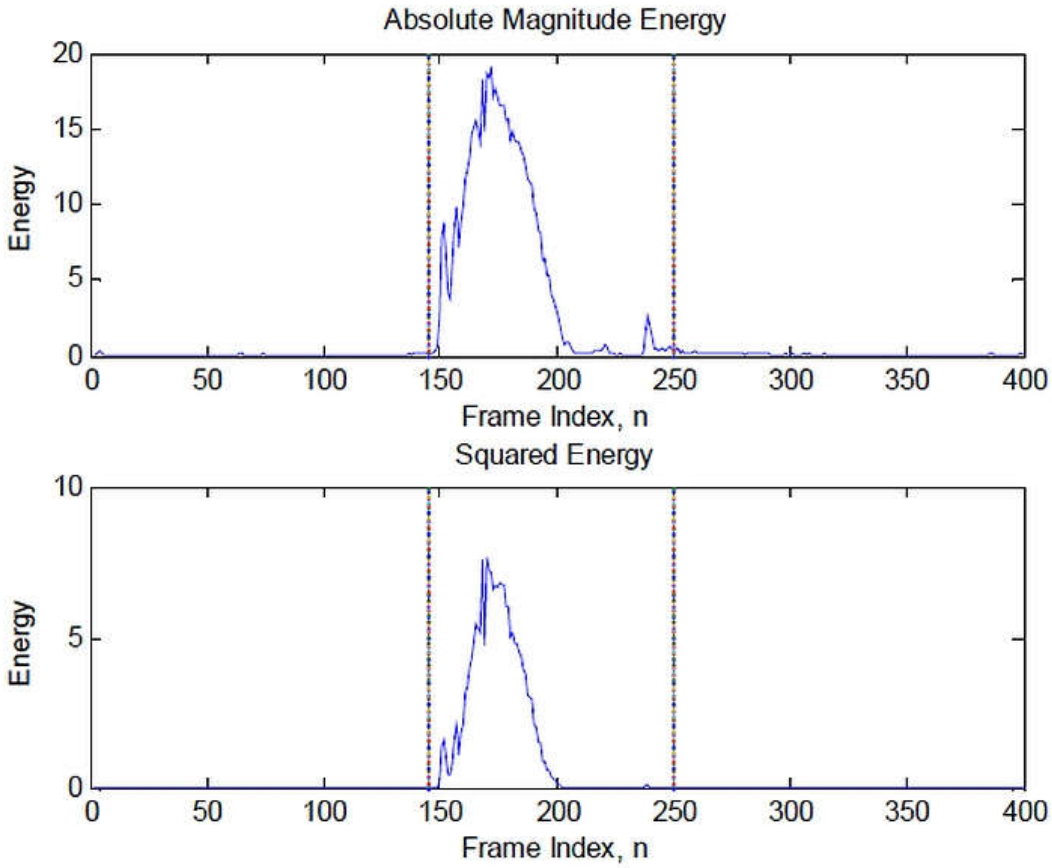


FIGURE 7.11: Absolute magnitude energy and Squared magnitude energy for word "tem"

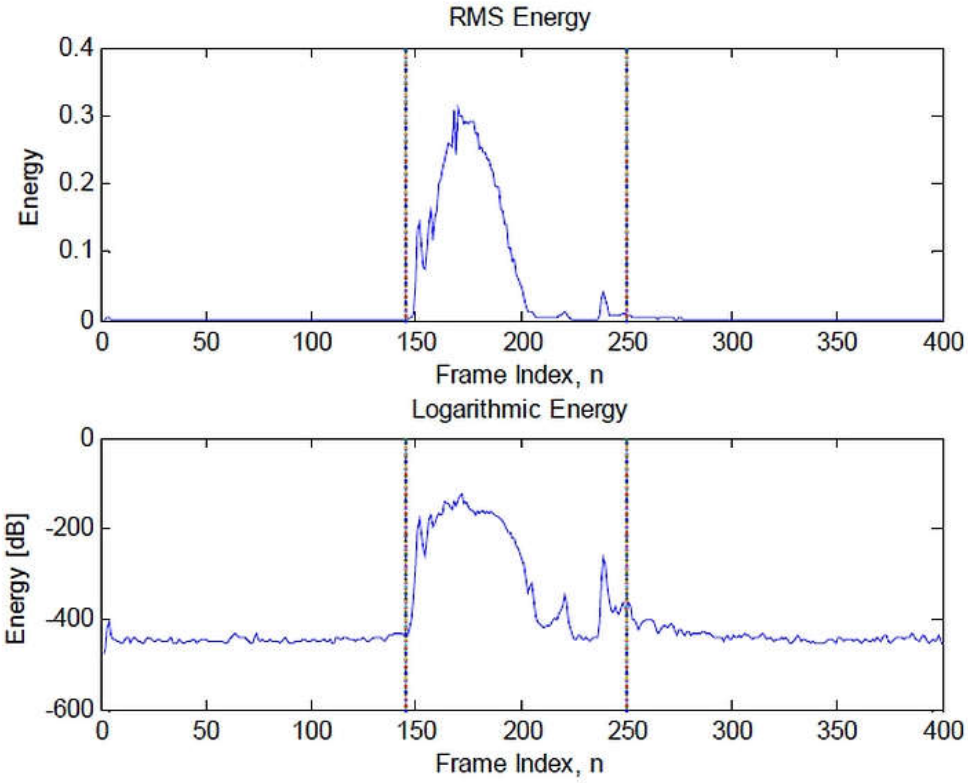


FIGURE 7.12: RMS energy and Logarithmic energy for word "tem"

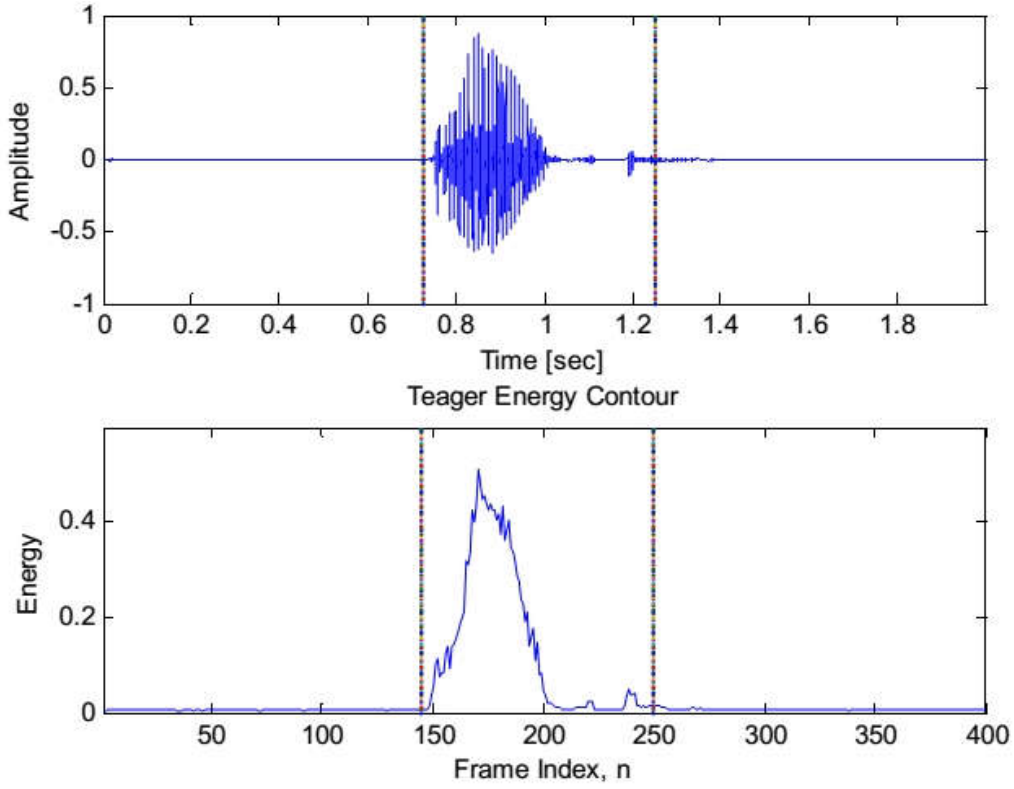


FIGURE 7.13: Speech Waveform and Teager Energy Plot.

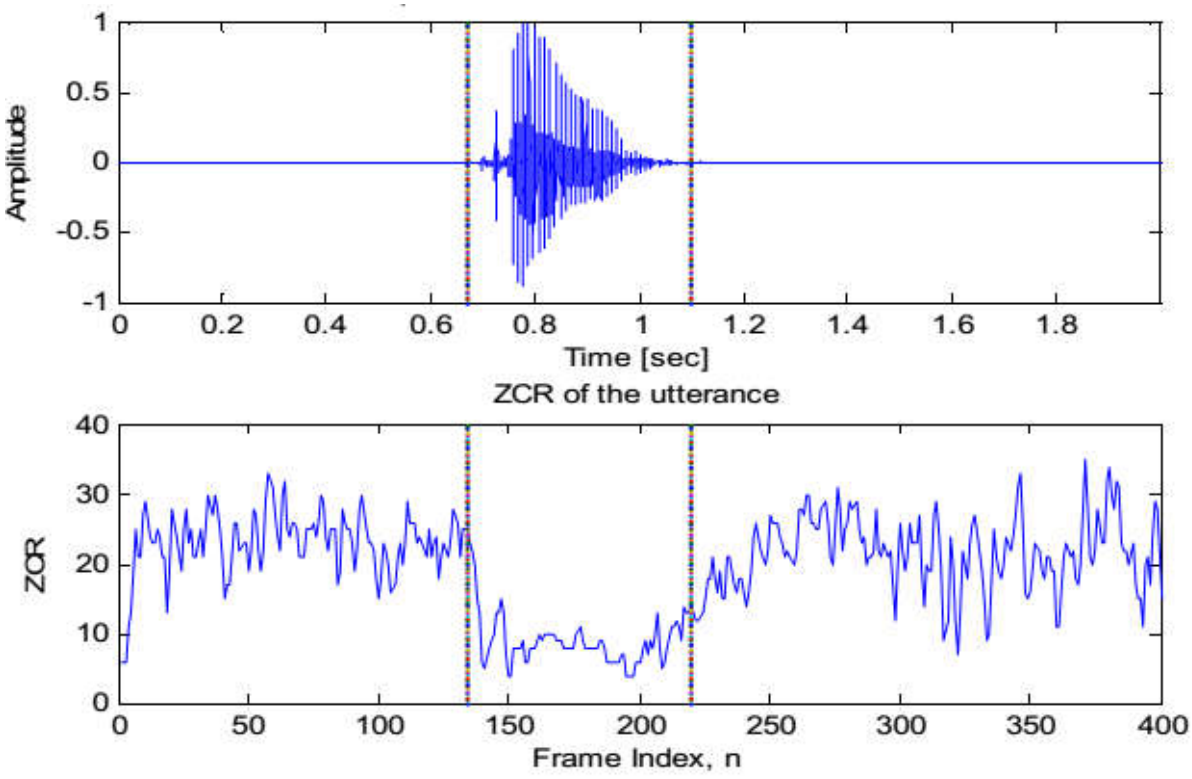


FIGURE 7.14: ZCR Plot for Noise Free Speech Signal.

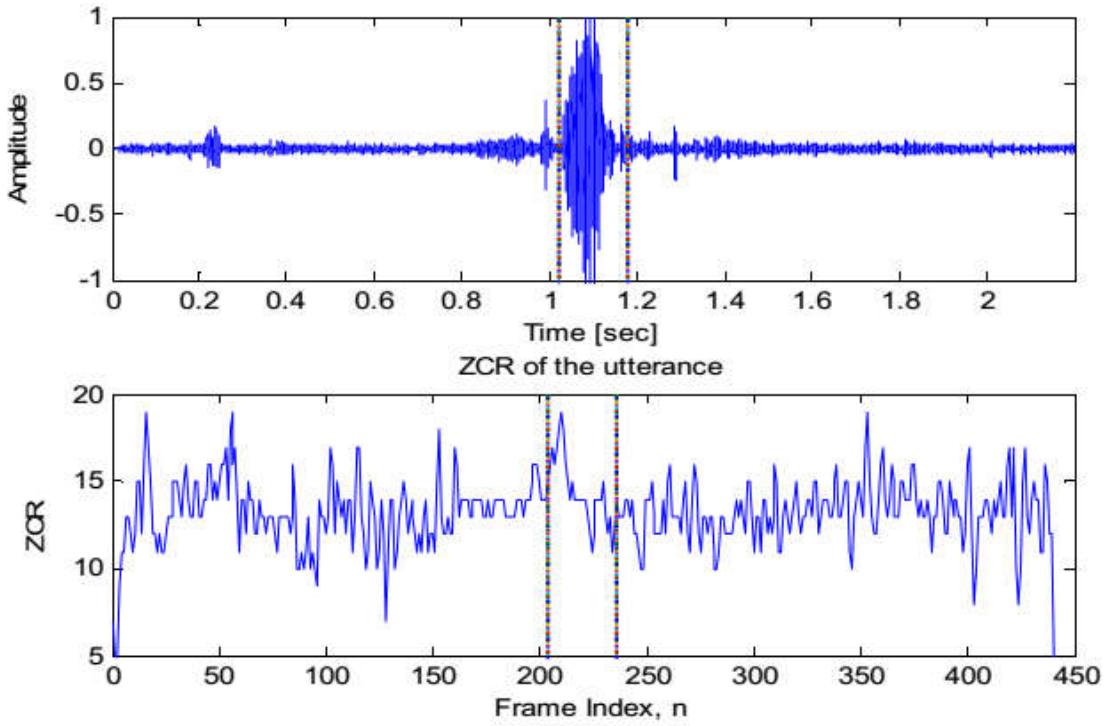


FIGURE 7.15: ZCR Plot for Noisy Speech Signal.

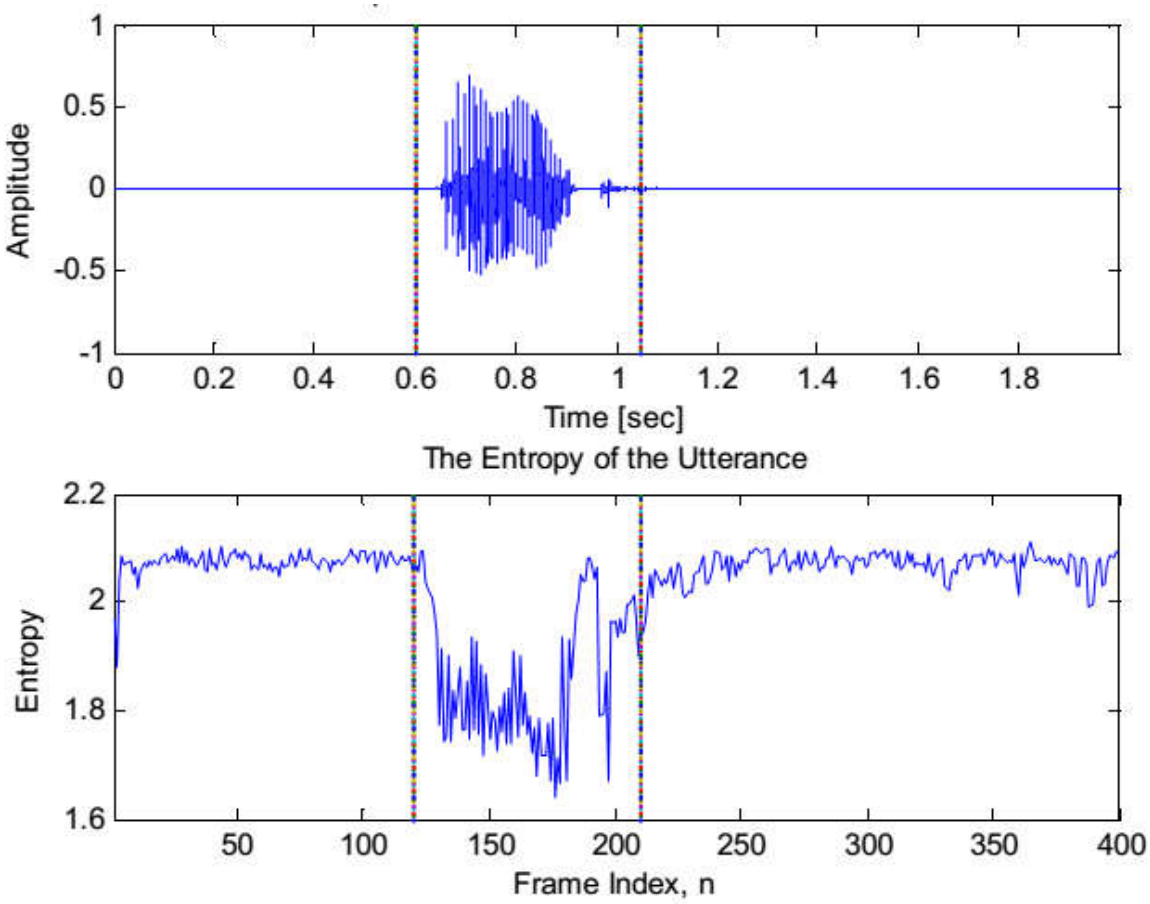


FIGURE 7.16: Entropy Feature Curve.

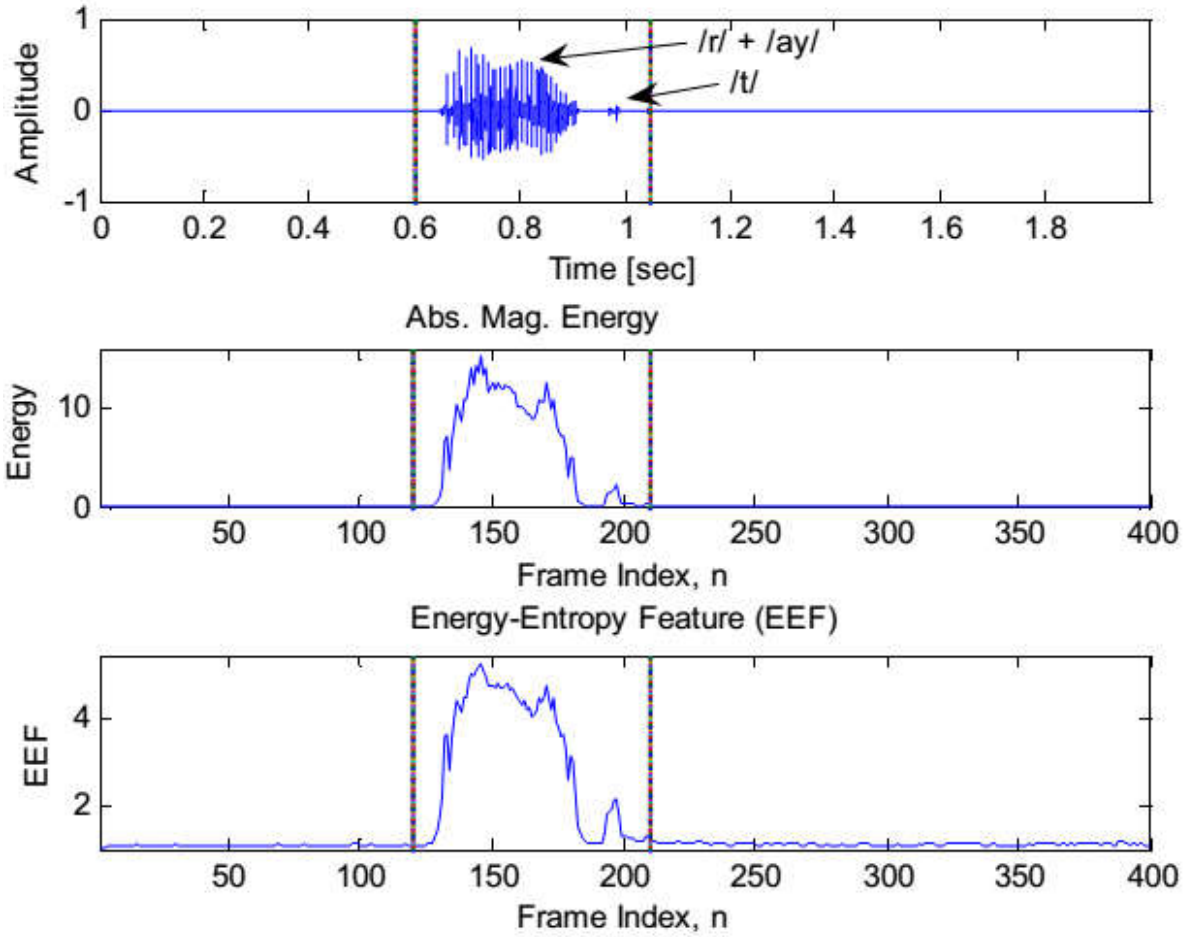


FIGURE 7.17: Absolute Magnitude Energy and Energy Entropy Curve.

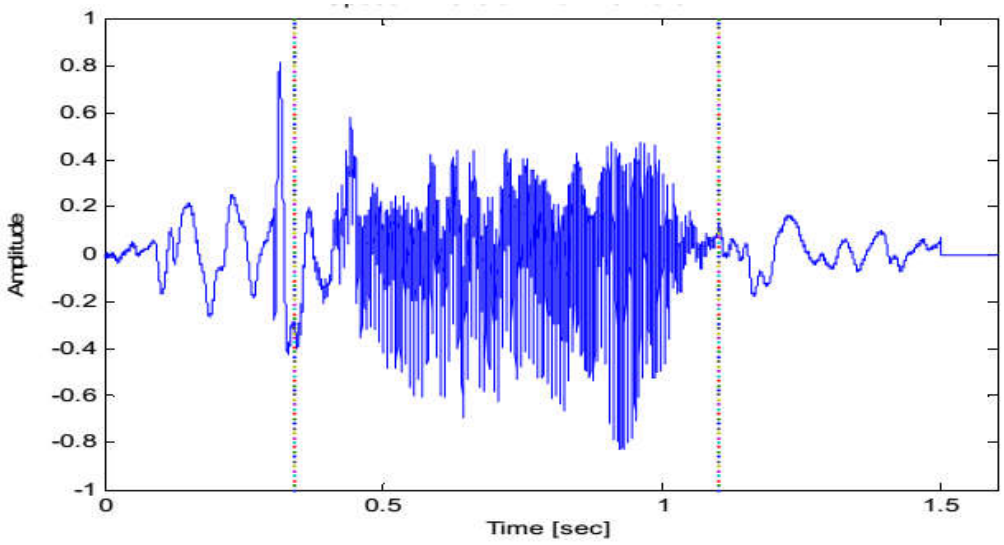


FIGURE 7.18: speech waveform for word "આમ"

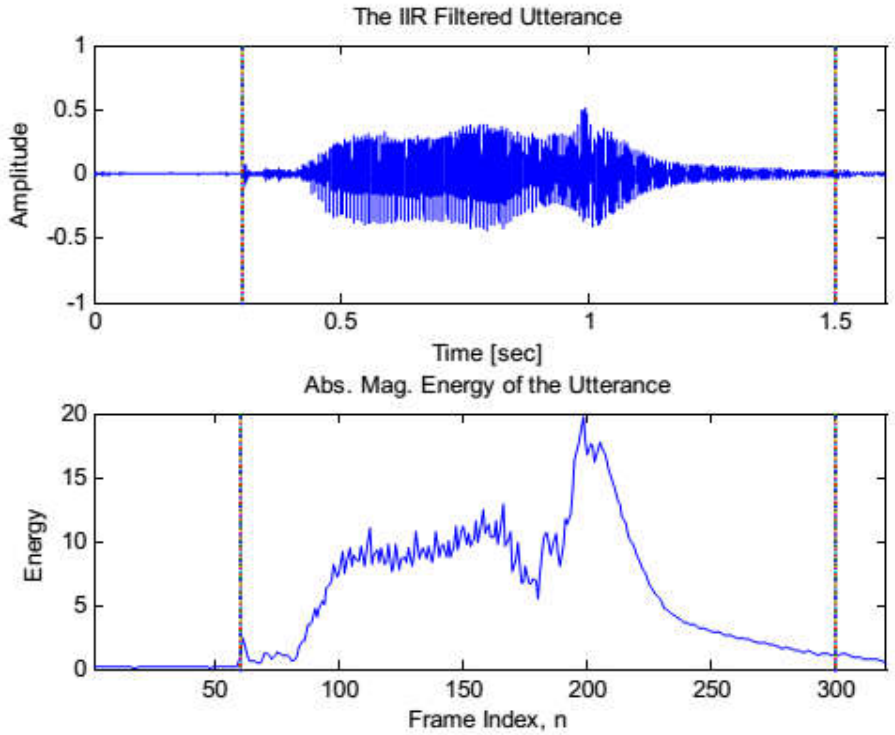


FIGURE 7.19: IIR filtered utterance for word "આમ" and corresponding absolute magnitude energy. Detected word boundary is indicated by the dotted line.

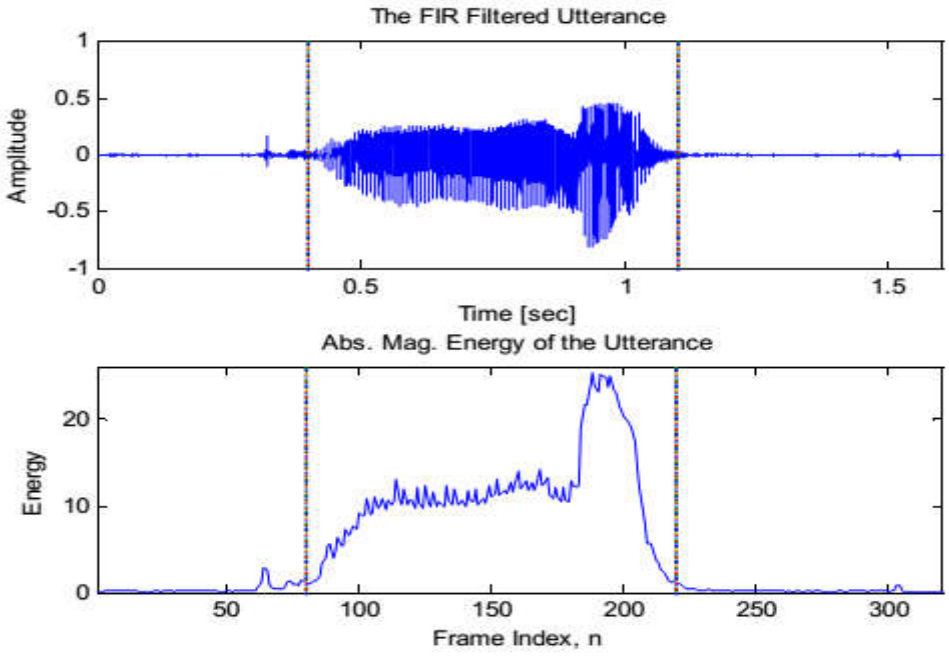


FIGURE 7.20: FIR filtered utterance for word "આમ" and corresponding absolute magnitude energy. Detected word boundary is indicated by the dotted line.

TABLE 7.2: Average recognition rates for Training data; (50 - 7) network configuration; MFCCs as input features

Overall Classification		DATA belong to:									
Rate = 84.72%		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**95.88**	0.28	3.88	0.26	0.28	0.00	0.58	0.52	0.25	0.25
	જમણી	0.35	**95.48**	2.46	2.30	2.03	0.66	0.05	0.25	0.45	0.44
	ઉપર	1.91	0.85	**89.48**	2.67	0.26	0.08	0.15	0.45	0.68	0.58
	નીચે	0.02	0.70	1.68	**89.10**	2.46	0.59	0.13	0.33	0.02	0.78
	આગળ	0.00	0.91	0.58	2.30	**89.11**	3.33	0.06	0.22	0.41	0.45
	પાછળ	0.15	0.76	0.17	1.33	3.66	**94.21**	0.10	0.75	0.36	0.02
	આજુ	0.70	0.03	0.76	1.04	1.21	0.14	**96.93**	2.45	0.48	0.21
	બાજુ	0.33	0.25	0.35	0.53	0.21	0.25	0.25	**95.23**	1.70	0.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	1.27	0.25	**92.25**	2.89
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	3.40	**94.25**

TABLE 7.3: Average recognition rates for Testing data; (50 - 7) network configuration; MFCCs as input features

Overall Classification		DATA belong to:									
Rate = 80.06%		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**94.30**	0.41	6.20	0.65	0.37	0.01	0.05	0.52	0.35	0.25
	જમણી	1.20	**91.63**	3.93	3.81	2.37	0.85	0.14	0.25	1.11	0.44
	ઉપર	2.65	0.58	**82.35**	5.02	0.36	0.07	0.72	0.45	0.68	0.58
	નીચે	0.11	1.49	2.35	**81.68**	3.97	2.04	0.48	1.33	0.02	0.78
	આગળ	0.01	1.84	1.44	3.46	**86.31**	3.08	0.19	3.22	0.49	0.65
	પાછળ	0.13	2.86	1.36	3.04	4.11	**92.64**	0.36	2.35	2.35	2.50
	આજુ	0.60	0.20	1.37	1.33	1.51	0.31	**96.06**	4.45	2.45	0.21
	બાજુ	0.33	0.25	0.35	0.53	0.21	0.25	2.25	**83.23**	1.70	2.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	0.27	3.25	**85.25**	4.89
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	1.05	5.60	**87.25**

TABLE 7.4: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર*" (50 - 7) network configuration; MFCCs as input features.

		DATA BELONGING TO:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	22.59	3.80
	જમણી	12.51	24.87
	ઉપર	9.22	2.85
	નીચે	12.99	53.74
	આગળ	7.85	2.33
	પાછળ	7.58	12.42
	આજુ	18.26	0.00
	બાજુ	2.33	0.25
	આમ	4.35	0.52
	તેમ	2.89	0.22

TABLE 7.5: Average recognition rates for Training data; (100 - 7) network configuration; MFCCs as input features

Overall Classification Rate = 88.68%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**97.25**	0.05	1.03	0.03	0.03	0.00	0.07	0.52	0.25	0.25
	જમણી	0.34	**98.81**	1.28	0.73	0.69	0.55	0.01	0.25	0.45	0.44
	ઉપર	0.48	0.28	**96.67**	1.09	0.03	0.00	0.03	0.45	0.68	0.58
	નીચે	0.00	0.20	0.63	**96.62**	1.65	0.88	0.06	0.33	0.02	0.78
	આગળ	0.00	0.35	0.07	0.75	**96.18**	0.11	0.01	0.22	0.41	0.45
	પાછળ	0.05	0.30	0.16	0.51	1.18	**98.44**	0.00	0.35	0.36	0.02
	આજુ	0.08	0.00	0.17	0.27	0.25	0.03	**99.82**	0.25	0.48	0.21
	બાજુ	0.33	0.25	0.35	0.43	0.21	0.25	0.25	**98.25**	0.25	0.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	0.84	0.25	**96.25**	0.45
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	0.40	**97.25**

TABLE 7.6: Average recognition rates for Testing data; (100 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 88.68%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**95.25**	2.05	1.03	0.03	0.03	0.00	0.07	0.52	0.25	0.25
	જમણી	1.34	**96.11**	1.28	0.73	0.69	0.55	0.01	0.25	0.45	0.44
	ઉપર	0.98	0.28	**96.17**	1.09	0.03	0.00	0.03	0.45	0.68	0.58
	નીચે	0.45	0.20	0.63	**96.02**	1.65	0.88	0.06	0.33	0.02	0.78
	આગળ	0.98	0.35	0.07	0.75	**94.18**	2.11	0.01	0.22	0.41	0.45
	પાછળ	0.05	0.30	0.16	0.51	2.18	**95.44**	0.00	0.35	0.36	0.02
	આજુ	0.08	0.00	0.17	0.27	0.25	0.03	**96.82**	0.25	0.48	0.21
	બાજુ	0.33	0.25	0.35	0.43	0.21	0.25	2.25	**97.65**	0.25	0.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	0.84	0.25	**96.95**	0.45
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	0.40	**97.25**

TABLE 7.7: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર*;" (100 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	35.25	3.98
	જમણી	11.82	38.55
	ઉપર	15.23	5.01
	નીચે	7.25	15.25
	આગળ	12.29	5.60
	પાછળ	6.28	13.00
	આજુ	2.15	8.25
	બાજુ	3.25	4.25
	આમ	4.35	3.26
	તેમ	2.89	3.25

TABLE 7.8: Average recognition rates for Training data; (150 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 88.5%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**97.45**	0.95	0.36	0.20	0.21	0.11	0.21	0.12	0.2	0.25
	જમણી	0.50	**97.52**	0.41	0.30	0.25	0.18	0.11	0.25	0.45	0.12
	ઉપર	0.16	0.12	**97.85**	0.21	0.12	0.12	0.32	0.21	0.23	0.1
	નીચે	0.20	0.65	0.32	**97.66**	0.21	0.31	0.21	0.33	0.2	0.2
	આગળ	0.32	0.10	0.14	0.45	**97.45**	0.68	0.14	0.22	0.41	0.3
	પાછળ	0.56	0.25	0.51	0.48	1.20	**98.45**	0.12	0.21	0.12	0.22
	આજુ	0.21	0.21	0.32	0.10	0.23	0.21	**97.45**	1.32	0.48	0.21
	બાજુ	0.33	0.41	0.12	0.21	0.56	0.25	1.21	**97.85**	0.12	0.1
	આમ	0.32	0.36	0.21	0.31	0.22	0.21	0.2	0.3	**97.14**	0.85
	તેમ	0.43	0.21	0.21	0.25	0.32	0.3	0.32	0.12	1.35	**98.45**

TABLE 7.9: Average recognition rates for Testing data; (150 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 83.9%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**92.56**	1.01	0.36	0.20	0.98	1.20	0.89	0.52	1.2	0.25
	જમણી	3.50	**92.65**	0.89	0.30	0.25	0.18	0.45	0.25	0.45	1.2
	ઉપર	0.16	1.20	**92.69**	2.30	0.87	0.14	0.65	0.45	0.68	0.95
	નીચે	1.20	0.65	1.20	**91.68**	1.35	0.31	0.21	0.33	1.25	0.78
	આગળ	0.32	0.98	1.04	0.45	**91.25**	3.25	1.20	0.22	0.41	0.45
	પાછળ	0.56	0.10	1.10	0.48	2.95	**93.25**	0.45	0.75	0.98	0.45
	આજુ	0.98	1.20	1.20	0.10	0.89	0.84	**92.36**	4.56	0.48	0.35
	બાજુ	0.33	0.88	0.55	2.23	0.56	0.25	3.35	**92.15**	1.7	0.98
	આમ	0.88	0.78	0.78	1.25	0.89	0.25	0.14	0.98	**91.12**	2.21
	તેમ	0.43	0.98	0.62	1.95	0.44	0.85	0.45	0.45	2.5	**93.21**

TABLE 7.10: Average recognition rates for the words "ડાબી બહાર" and "જમણી બહાર;" (150 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	25.58	4.22
	જમણી	10.94	46.95
	ઉપર	8.55	6.04
	નીચે	19.52	22.25
	આગળ	3.25	3.39
	પાછળ	7.20	12.06
	આજુ	15.56	0.18
	બાજુ	2.33	0.25
	આમ	4.35	3.25
	તેમ	2.89	2.22

TABLE 7.11: Average recognition rates for Training data; (150 - 7) network configuration; RCs as input features.

Overall Classification Rate = 83.02%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**92.84**	2.45	2.69	0.98	0.13	0.02	0.31	0.52	0.35	0.56
	જમણી	2.45	**91.84**	1.25	0.45	0.71	1.19	1.25	0.25	0.39	0.89
	ઉપર	0.68	0.25	**91.25**	0.58	0.98	0.45	1.10	0.45	0.68	0.58
	નીચે	0.98	0.15	1.40	**90.84**	0.45	1.20	0.53	0.35	0.55	0.78
	આગળ	0.45	0.14	0.45	1.20	**91.25**	3.25	0.93	0.56	0.49	0.65
	પાછળ	0.35	0.55	0.24	1.43	3.45	**90.45**	0.45	1.45	0.84	0.56
	આજુ	0.98	1.32	1.25	1.23	1.25	0.88	**91.25**	3.20	0.56	0.21
	બાજુ	0.55	2.25	0.35	1.25	1.02	0.56	3.35	**91.35**	1.20	0.56
	આમ	0.85	1.20	0.85	1.34	0.68	1.20	0.89	0.85	**91.56**	4.89
	તેમ	0.36	0.22	0.62	0.89	0.79	0.85	0.47	1.20	3.65	**90.62**

TABLE 7.12: Average recognition rates for Testing data; (150 - 7) network configuration; RCs as input features.

Overall Classification Rate = 76.29%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**88.25**	0.41	5.46	0.29	0.29	2.2	0.7	0.52	0.35	0.25
	જમણી	1.48	**86.43**	4.96	3.29	4.66	3.32	0.64	0.25	1.11	0.44
	ઉપર	4.22	2.88	**80.73**	3.8	1.43	0.22	1.84	0.45	0.68	0.58
	નીચે	0.23	1.89	3.22	**78.3**	9.11	2.35	1.04	1.33	0.45	0.78
	આગળ	0.48	1.24	1.55	10.12	**80.19**	4.25	0.85	3.22	1.89	0.65
	પાછળ	1.2	4.68	1.05	2.37	3.2	**80.45**	0.42	2.35	2.35	2.5
	આજુ	0.78	0.85	2.11	0.84	0.55	1.25	**87.45**	4.45	2.45	0.21
	બાજુ	1.22	1.23	0.56	0.25	0.25	1.86	5.25	**86.25**	2.2	2.45
	આમ	0.45	0.55	0.45	1.34	0.15	3.45	2.4	1.2	**83.85**	4.89
	તેમ	1.85	0.21	0.21	0.25	0.79	1.51	0.23	0.89	4.98	**87.25**

TABLE 7.13: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર*;" (150 - 7) network configuration; RCs as input features

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	25.25	21.25
	જમણી	18.25	35.25
	ઉપર	7.25	6.99
	નીચે	8.25	12.26
	આગળ	5.25	5.25
	પાછળ	4.91	3.25
	આજુ	14.48	4.85
	બાજુ	5.56	4.45
	આમ	7.25	3.25
	તેમ	3.85	3.35

TABLE 7.14: Average recognition rates for Training data; (30-20 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 87.68%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**90.21**	3.38	1.00	1.23	1.25	1.20	0.98	0.52	0.95	0.85
	જમણી	2.35	**89.45**	0.98	0.98	0.70	0.95	1.23	0.89	0.45	0.98
	ઉપર	0.57	1.25	**88.65**	1.56	1.25	0.89	0.45	0.45	0.68	1.25
	નીચે	1.25	0.40	0.85	**88.21**	1.16	1.95	2.32	0.33	0.86	1.40
	આગળ	0.25	0.52	1.25	1.11	**86.25**	3.25	0.32	0.22	0.41	0.89
	પાછળ	1.50	0.53	0.89	0.87	4.25	**88.55**	0.02	0.75	1.20	1.20
	આજુ	2.20	2.20	1.24	0.77	1.12	1.20	**88.85**	5.25	0.48	0.55
	બાજુ	0.45	0.85	1.35	1.20	2.32	1.25	4.55	**87.56**	0.52	0.45
	આમ	0.36	1.25	2.65	3.20	1.21	0.32	0.25	1.20	**90.20**	3.25
	તેમ	0.89	0.45	1.20	1.20	0.98	0.98	1.25	3.20	4.55	**89.55**

TABLE 7.15: Average recognition rates for Testing data; (30-20 - 7) network configuration; MFCCs as input features

Overall Classification Rate = 79.219%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**89.45**	4.52	2.85	0.36	0.11	0.03	0.57	0.52	0.85	0.45
	જમણી	3.52	**89.25**	3.35	0.45	1.15	1.90	0.45	0.25	0.65	0.85
	ઉપર	0.45	1.17	**89.25**	3.68	0.23	0.04	0.50	0.45	0.68	0.68
	નીચે	0.89	0.88	1.52	**91.25**	3.02	1.85	0.63	0.33	0.85	0.52
	આગળ	0.84	0.48	0.77	0.85	**89.45**	4.56	1.36	0.65	0.98	0.69
	પાછળ	1.12	0.86	0.64	0.35	4.98	**89.85**	0.41	1.65	0.35	1.10
	આજુ	0.17	0.52	0.64	0.54	0.69	0.14	**90.45**	4.45	0.84	0.21
	બાજુ	1.23	0.85	0.56	0.85	0.21	0.54	4.50	**89.95**	1.70	2.45
	આમ	1.32	0.25	0.84	0.78	0.35	0.52	0.45	1.32	**89.25**	4.25
	તેમ	**89.45**	4.52	2.85	0.36	0.11	0.03	0.57	0.52	0.85	0.45

TABLE 7.16: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર*" (30-20 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	35.25	4.86
	જમણી	4.45	50.25
	ઉપર	5.50	6.30
	નીચે	6.85	12.25
	આગળ	11.89	2.80
	પાછળ	6.44	12.83
	આજુ	12.25	4.25
	બાજુ	6.85	3.25
	આમ	4.35	1.85
	તેમ	6.25	2.25

TABLE 7.17: Average recognition rates for Training data; (40-20 - 7) network configuration; MFCCs as input features

Overall Classification Rate = 87.42%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**95.88**	0.28	0.85	0.26	0.28	0	0.58	0.52	0.25	0.25
	જમણી	0.35	**95.48**	0.25	0.45	0.4	0.66	0.05	0.25	0.45	0.44
	ઉપર	1.91	0.85	**96.25**	0.23	0.26	0.08	0.15	0.45	0.68	0.58
	નીચે	0.02	0.7	0.68	**96.3**	0.23	0.59	0.13	0.33	0.02	0.78
	આગળ	0	0.91	0.58	0.12	**95.45**	1.85	0.06	0.22	0.41	0.45
	પાછળ	0.15	0.76	0.17	1.33	2.25	**96.21**	0.1	0.75	0.36	0.02
	આજુ	0.7	0.03	0.76	1.04	0.32	0.14	**96.93**	1.45	0.48	0.21
	બાજુ	0.33	0.25	0.35	0.53	0.21	0.25	1.32	**96.23**	0.25	0.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	0.24	0.25	**96.65**	1.25
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	1.02	**96.25**

TABLE 7.18: Average recognition rates for Testing data; (40-20 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 83.62%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**90.35**	1.95	1.20	0.65	0.37	0.01	0.85	0.52	0.35	0.25
	જમણી	4.25	**91.63**	1.21	1.10	2.37	0.85	0.66	0.25	0.12	0.44
	ઉપર	2.65	0.58	**92.35**	1.23	0.36	0.07	0.72	0.45	0.68	0.58
	નીચે	0.11	1.49	0.45	**91.65**	0.85	2.04	0.48	0.33	0.02	0.78
	આગળ	0.45	1.84	1.44	2.10	**91.54**	3.08	0.62	0.85	0.49	0.65
	પાછળ	0.59	1.68	1.36	1.20	4.11	**92.64**	0.36	0.45	0.45	0.45
	આજુ	0.60	0.20	1.37	1.33	0.12	0.31	**91.25**	3.25	0.24	0.21
	બાજુ	0.33	0.25	0.35	0.53	0.21	0.25	4.56	**94.25**	0.32	0.23
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	0.27	0.21	**92.24**	3.95
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	5.60	**92.54**

TABLE 7.19: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર* (40-20 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	35.25	4.86
	જમણી	4.45	50.25
	ઉપર	5.50	6.30
	નીચે	6.85	12.25
	આગળ	11.89	2.80
	પાછળ	6.44	12.83
	આજુ	12.25	4.25
	બાજુ	6.85	3.25
	આમ	4.35	1.85
	તેમ	6.25	2.25

TABLE 7.20: Average recognition rates for Training data; (50-30 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 89.01%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**98.65**	1.32	0.32	0.25	0.25	1.25	0.22	0.52	0.25	0.25
	જમણી	1.20	**98.85**	0.52	0.88	0.45	0.45	0.25	0.25	0.45	0.44
	ઉપર	0.25	0.25	**97.88**	0.35	0.36	0.55	0.32	0.45	0.68	0.58
	નીચે	0.45	0.22	0.25	**98.65**	0.15	0.42	0.12	0.33	0.02	0.78
	આગળ	0.45	0.42	0.88	0.45	**97.45**	2.52	0.22	0.22	0.41	0.45
	પાછળ	0.49	0.34	0.45	0.25	2.35	**97.58**	0.15	0.75	0.36	0.02
	આજુ	0.56	0.48	0.22	0.11	0.30	0.21	**97.23**	2.45	0.48	0.21
	બાજુ	0.12	0.25	0.35	0.65	0.21	0.36	1.52	**97.58**	0.12	0.45
	આમ	0.58	0.35	0.36	0.36	0.35	0.45	0.24	0.25	**97.56**	2.89
	તેમ	0.12	0.15	0.85	0.58	0.44	0.25	0.33	0.05	1.25	**97.66**

TABLE 7.21: Average recognition rates for Testing data; (50-30- 7) network configuration; MFCCs as input features.

Overall Classification Rate = 84.12%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**93.65**	3.20	1.12	1.25	0.68	0.03	0.85	0.52	0.35	0.85
	જમણી	3.25	**93.25**	1.23	0.54	0.48	1.35	0.56	0.56	0.85	0.44
	ઉપર	0.25	0.45	**93.54**	0.85	0.18	1.23	0.89	0.65	0.68	0.58
	નીચે	0.21	0.56	2.69	**93.54**	0.41	0.85	0.56	0.25	0.45	0.78
	આગળ	2.14	0.66	0.89	93.55	**91.66**	4.25	0.98	0.54	0.49	0.56
	પાછળ	0.12	0.98	1.10	1.57	4.23	**92.25**	0.30	0.96	0.56	0.84
	આજુ	0.23	0.25	0.35	0.36	1.35	0.54	**92.25**	4.45	0.65	0.52
	બાજુ	0.33	1.21	0.35	0.89	0.35	2.50	4.65	**91.32**	0.55	0.45
	આમ	2.25	1.85	0.85	0.66	0.89	1.35	0.25	0.25	**92.25**	3.56
	તેમ	1.25	0.87	0.62	0.25	0.55	1.25	0.85	0.52	4.56	**91.74**

TABLE 7.22: Average recognition rates for the words "ડાબી બહાર" and "જમણી બહાર" (50-30 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	45.25	4.89
	જમણી	9.21	44.12
	ઉપર	6.45	8.82
	નીચે	12.25	12.55
	આગળ	11.25	3.67
	પાછળ	3.25	10.25
	આજુ	4.25	2.25
	બાજુ	1.32	3.54
	આમ	4.35	4.56
	તેમ	2.89	5.86

TABLE 7.23: Average recognition rates for Training data; (60-40 - 7) network configuration; MFCCs as input features.

Overall Classification Rate = 87.16%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**96.35**	1.32	0.32	0.25	0.25	1.25	0.22	0.52	0.25	0.25
	જમણી	1.20	**96.85**	0.52	0.88	0.45	0.45	0.25	0.25	0.45	0.44
	ઉપર	0.25	0.25	**96.20**	0.35	0.36	0.55	0.32	0.45	0.68	0.58
	નીચે	0.45	0.22	0.25	**96.56**	0.15	0.42	0.12	0.33	0.02	0.78
	આગળ	0.45	0.42	0.88	0.45	**95.21**	2.52	0.22	0.22	0.41	0.45
	પાછળ	0.49	0.34	0.45	0.25	2.35	**94.45**	0.15	0.75	0.36	0.02
	આજુ	0.56	0.48	0.22	0.11	0.30	0.21	**97.23**	2.45	0.48	0.21
	બાજુ	0.12	0.25	0.35	0.65	0.21	0.36	1.52	**95.23**	0.12	0.45
	આમ	0.58	0.35	0.36	0.36	0.35	0.45	0.24	0.25	**96.45**	2.89
	તેમ	0.12	0.15	0.85	0.58	0.44	0.25	0.33	0.05	1.25	**94.25**

TABLE 7.24: Average recognition rates for Testing data; (60-40- 7) network configuration; MFCCs as input features.

Overall Classification Rate = 82.7%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**90.65**	3.20	1.12	1.25	0.68	0.03	0.85	0.52	0.35	0.85
	જમણી	3.25	**90.32**	1.23	0.54	0.48	1.35	0.56	0.56	0.85	0.44
	ઉપર	0.25	0.45	**91.24**	0.85	0.18	1.23	0.89	0.65	0.68	0.58
	નીચે	0.21	0.56	2.69	**90.12**	0.41	0.85	0.56	0.25	0.45	0.78
	આગળ	2.14	0.66	0.89	3.64	**91.66**	4.25	0.98	0.54	0.49	0.56
	પાછળ	0.12	0.98	1.10	1.57	4.23	**90.51**	0.30	0.96	0.56	0.84
	આજુ	0.23	0.25	0.35	0.36	1.35	0.54	**90.45**	4.45	0.65	0.52
	બાજુ	0.33	1.21	0.35	0.89	0.35	2.50	4.65	**91.32**	0.55	0.45
	આમ	2.25	1.85	0.85	0.66	0.89	1.35	0.25	0.25	**91.66**	3.56
	તેમ	1.25	0.87	0.62	0.25	0.55	1.25	0.85	0.52	4.56	**91.74**

TABLE 7.25: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર* (60-40 - 7) network configuration; MFCCs as input features.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	45.25	3.80
	જમણી	5.25	55.85
	ઉપર	3.20	2.85
	નીચે	4.50	12.25
	આગળ	3.25	2.33
	પાછળ	5.24	10.21
	આજુ	18.26	1.20
	બાજુ	6.50	3.20
	આમ	5.62	4.50
	તેમ	2.98	4.50

TABLE 7.26: Average recognition rates for Training data; (60-40 - 7) network configuration; RCs as input features.

Overall Classification Rate = 82.92%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**91.25**	2.98	1.40	1.25	0.45	0.52	1.32	0.52	0.95	1.24
	જમણી	2.35	**90.45**	0.45	2.12	0.77	0.85	2.35	0.45	0.45	0.89
	ઉપર	1.25	1.20	**91.25**	0.39	0.38	0.95	1.85	1.01	0.68	0.58
	નીચે	1.55	0.18	2.20	**90.15**	0.78	0.45	0.65	0.33	1.20	0.78
	આગળ	0.55	0.86	1.39	0.58	**91.25**	3.85	0.45	0.22	2.40	0.45
	પાછળ	0.41	0.22	0.38	0.19	4.52	**91.68**	1.20	0.75	1.20	0.45
	આજુ	0.32	0.48	0.10	1.52	2.30	1.20	**91.54**	4.50	2.87	1.25
	બાજુ	1.85	0.50	1.30	1.89	0.89	0.68	4.50	**91.25**	1.70	2.30
	આમ	2.20	2.30	2.50	2.25	0.35	2.30	1.27	2.30	**91.85**	4.56
	તેમ	0.43	2.98	1.50	1.68	1.40	1.98	3.20	0.98	4.52	**91.45**

TABLE 7.27: Average recognition rates for Testing data; (60-40- 7) network configuration; RCs as input features

Overall Classification Rate = 76.12%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**84.58**	0.56	5.36	0.26	0.28	1.32	3.45	0.52	2.32	0.25
	જમણી	5.20	**83.45**	3.88	2.84	3.36	3.25	2.89	0.25	1.11	0.44
	ઉપર	3.25	3.61	**81.40**	3.41	1.83	0.42	2.33	0.45	1.55	0.58
	નીચે	1.20	3.00	3.68	**83.21**	9.81	1.94	1.46	1.33	2.05	0.78
	આગળ	0.68	1.53	2.71	10.12	**83.45**	5.25	1.09	3.22	1.58	0.65
	પાછળ	0.39	1.21	1.08	2.37	2.61	**84.22**	0.89	4.25	2.35	2.50
	આજુ	1.69	0.78	1.87	1.39	0.89	2.30	**83.12**	4.45	2.45	1.25
	બાજુ	2.20	0.25	0.35	0.53	0.21	0.25	4.58	**83.45**	2.35	2.45
	આમ	1.22	0.98	0.02	1.25	1.20	0.52	1.85	3.20	**83.56**	4.89
	તેમ	**84.58**	0.56	5.36	0.26	0.28	1.32	3.45	0.52	2.32	0.25

TABLE 7.28: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર* (60-40 - 7) network configuration; RCs as input features.

.		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	45.35	8.99
	જમણી	11.68	41.20
	ઉપર	12.00	3.22
	નીચે	3.55	13.55
	આગળ	11.25	6.55
	પાછળ	5.07	5.56
	આજુ	6.40	9.45
	બાજુ	1.32	1.25
	આમ	2.22	4.56
	તેમ	1.55	5.86

TABLE 7.29: Average recognition rates for Training data; (40-20 - 7) network configuration; MFCCs as input features. with LM algorithm

Overall Classification Rate = 87.02%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**95.88**	0.28	0.89	0.26	0.28	0	0.58	0.52	0.25	0.25
	જમણી	0.35	**95.48**	1.23	0.45	0.35	0.66	0.05	0.25	0.45	0.44
	ઉપર	1.91	0.85	**94.45**	1.65	0.26	0.08	0.15	0.45	0.68	0.58
	નીચે	0.02	0.7	1.68	**95.25**	0.45	0.59	0.13	0.33	0.02	0.78
	આગળ	0	0.91	0.58	0.3	**95.65**	1.33	0.06	0.22	0.41	0.45
	પાછળ	0.15	0.76	0.17	0.85	2.68	**96.25**	0.1	0.75	0.36	0.02
	આજુ	0.7	0.03	0.76	0.85	0.21	0.14	**96.93**	1.45	0.48	0.21
	બાજુ	0.33	0.25	0.35	0.53	0.21	0.25	1.25	**96.45**	0.98	0.45
	આમ	0.24	0.52	0.02	0.25	0.35	0.52	1.27	0.25	**95.25**	1.85
	તેમ	0.43	0.22	0.62	0.25	0.44	0.23	0.47	0.05	1.25	**95.65**

TABLE 7.30: Average recognition rates for Testing data; (40-20 - 7) network configuration; MFCCs as input features. with LM algorithm.

Overall Classification Rate = 78.74%		DATA belong to:									
		ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ
DATA identified as	ડાબી	**86.15**	3.12	2.32	0.65	0.37	1.20	1.25	0.52	0.35	0.25
	જમણી	4.98	**86.49**	4.56	3.81	2.37	0.85	0.95	0.25	1.11	0.44
	ઉપર	3.12	0.89	**86.45**	5.65	0.36	1.20	1.20	0.45	0.68	0.58
	નીચે	0.85	0.49	2.35	**85.12**	3.97	2.04	1.20	1.33	0.02	0.78
	આગળ	0.86	1.84	1.44	1.52	**86.31**	3.08	0.88	2.10	0.49	0.65
	પાછળ	0.85	1.85	1.36	1.25	4.11	**86.45**	0.95	1.35	1.30	2.50
	આજુ	0.95	1.20	1.37	1.33	1.85	0.98	**86.45**	4.45	1.45	0.98
	બાજુ	0.88	2.20	0.35	0.53	0.21	1.32	5.32	**87.65**	1.86	2.45
	આમ	0.69	1.20	0.02	0.25	0.35	1.56	0.89	1.25	**87.85**	4.89
	તેમ	0.85	0.86	0.62	0.25	0.44	1.40	1.25	0.85	5.60	**87.25**

TABLE 7.31: Average recognition rates for the words "*ડાબી બહાર*" and "*જમણી બહાર* (40-20 - 7) network configuration; MFCCs as input features. with LM algorithm.

		DATA belong to:	
		ડાબી બહાર	જમણી બહાર
DATA identified as	ડાબી	35.25	4.86
	જમણી	4.45	50.25
	ઉપર	5.50	6.30
	નીચે	6.85	12.25
	આગળ	11.89	2.80
	પાછળ	6.44	12.83
	આજુ	12.25	4.25
	બાજુ	6.85	3.25
	આમ	4.35	1.85
	તેમ	6.25	2.25

TABLE 7.32: overall classification rates for testing sets of all configurations.

Net-work Configu-ration	Vocabulary Words										Overall Avg. classify-cation rate
	ડાબી	જમણી	ઉપર	નીચે	આગળ	પાછળ	આજુ	બાજુ	આમ	તેમ	
50-10	81.45 (74.56, 88.34)	78.45 (71.25, 85.65)	82.35 (84.56, 80.14)	84.56 (86.25, 82.87)	78.18 (71.25, 85.11)	77.25 (82.56, 71.94)	76.65 (77.45, 75.85)	83.25 (85.36, 81.14)	81.25 (84.55, 77.95	84.25 (87.26, 81.24)	80.76 (84.56, 77.25)
100-10	83.45 (80.25, 86.65)	80.25 (72.56, 87.94)	85.56 (81.25, 89.87)	84.45 (84.22, 84.68)	80.22 (74.25, 86.19)	79.85 (71.62, 88.08)	82.56 (79.85, 85.27)	84.85 (80.25, 89.45)	83.56 (85.25, 81.87)	84.66 (84.56, 84.76)	82.84 (77.56, 86.66)
150-10	84.25 (81.52, 86.98)	82.25 (74.56, 89.94)	86.25 (82.25, 90.25)	88.56 (83.62, 93.5)	82.25 (76.25, 88.25)	81.55 (73.56, 89.54)	82.98 (78.56, 87.4)	86.25 (81.22, 91.28)	84.56 (86.56, 82.56)	86.56 (83.69, 89.43)	84.29 (80.45, 88.56)
150-10 (RC)	81.25 (84.52, 77.98)	73.25 (80.25, 66.25)	76.41 (78.45, 74.37)	74.52 (76.56, 72.48)	72.52 (69.85, 75.19)	73.25 (76.25, 70.25)	78.65 (76.56, 80.74)	80.65 (81.25, 80.05)	81.25 (83.25, 79.25)	77.85 (79.25, 76.45)	76.73 (72.29, 81.25)
30-20-10	82.52 (83.62, 81.42)	73.55 (70.25, 76.85)	83.25 (81.25, 85.25)	83.56 (78.95, 88.17)	78.53 (70.25, 86.81)	79.85 (70.25, 89.45)	82.25 (80.25, 84.25)	83.25 (84.25, 82.25)	83.45 (85.55, 81.35)	82.55 (84.25, 80.85)	81.27 (73.55, 83.45)
40-20-10	85.45 (84.25, 86.65)	76.86 (73.25, 80.47)	87.5 (85.25, 89.75)	85.45 (83.55, 87.35)	80.56 (74.65, 86.47)	81.56 (79.56, 83.56)	83.66 (81.25, 86.07)	86.52 (84.65, 88.39)	82.25 (84.33, 80.17)	84.58 (86.66, 82.5)	83.43 (76.86, 87.5)
50-30-10	87.62 (84.25, 90.99)	77.98 (70.25, 85.71)	88.45 (85.95, 90.95)	86.45 (84.58, 88.32)	76.85 (71.41, 82.29)	83.25 (77.36, 89.14)	84.65 (83.96, 85.34)	87.45 (86.78, 88.12)	83.56 (84.52, 82.6)	85.25 (84.14, 86.16)	84.15 (76.85, 88.45)
60-40-10	81.56 (79.56, 83.56)	76.85 (71.25, 82.45)	84.35(82.45,86.25)	89.15 (87.25, 91.05)	77.89 (72.56, 83.22)	79.88 (75.56, 84.2)	86.55 (84.69, 88.41)	83.25 (81.26, 85.24)	85.45 (83.25, 87.65)	80.56 (78.96, 82.16)	82.54 (76.85, 89.15)
60-40-10 (RC)	74.25 (72.25, 76.25)	74.55 (70.22, 78.88)	79.12 (78.12, 80.12)	78.65 (76.66, 80.64)	72.52 (68.54, 76.5)	72.52 (68.98, 76.06)	78.56 (77.11, 80.01)	77.45 (76.36, 78.54)	78.45 (76.68, 80.22)	79.55 (77.98, 81.12)	76.56 (72.52, 79.55)
40-20-10	80.11 (77.65, 82.57)	77.89(71.32,84.46)	80.44(77.89,82.99)	77.55 (75.65, 79.45)	77.85 (70.56, 85.14)	73.55 (67.52, 79. 58)	79.45 (76.98, 81.92)	76.66 (74.65, 78.67)	78.45 (73.12, 83.78)	80.15 (77.12, 83.18)	78.21 (73.55, 80.44)

CHAPTER 8

Conclusion and Future Work

8.1 Conclusion

Different configurations of neural networks are tested for the database created using in-ear microphone. From the results, we can conclude the following points:

From the results, we can conclude that:

- Results shows the great improvement in the accuracy for the in-ear microphone recording scheme compare to conventional microphone recording system.
- The spectrogram for two versions of the same word, i.e in-ear microphone recording and conventional microphone recording, shows that both waveforms are quite different from each other.
- Speech signal collected from in-ear microphone shows that higher frequency components are damped due to structure of the ear of human body.
- The waveform also shows that low frequency hum goes up to 100 Hz, for in-ear microphone recording, which may go up to 1.25 KHz for the conventional microphone recording system.
- In-ear microphone speech signals are more immune to external noise compare to conventional microphone system.
- Two level threshold mechanism of end-point detection algorithm able to separate silence from the word and able to find word boundary perfectly.
- Spoken words are identified correctly using different combinations of neural network.
- Overall average classification rates increase for two-layer and three-layer network configurations as the number of neurons in the hidden layers increase.

- The best recognition performance is obtained with 150-10 network for two-layer configuration and with 60-40-40 for the three-layer network.
- There is about an 8% difference between the average classification rates obtained with the best network structures, i.e., the (150-10) and (60-40-10) networks, using the MFCCs and RCs as features.
- Performance degradation is more severe and noticeable for the words આગળ, પાછળ, and જમણી.
- For this words, the 95% average classification rates confidence interval lower bounds for the network with RC coefficients goes down to 66.25%.
- We also observed that confidence interval spreads in case of RC are much larger than those obtained with the MFCCs, sometimes 8% to 14% for the words આગળ, પાછળ, and જમણી.
- MFCCs lead to better recognition rates than RCs.
- The highest recognition results are obtained with આમ, તેમ, ઉપર, નીચે, આજુ, બાજુ respectively.
- The performance difference between the networks trained using the CG or the LM scheme is also an important point to note.
- The network trained using the CG algorithm yields around a 3.5 percent higher recognition result than that obtained with the LM algorithm on the same network configuration.
- 5. 95% Confidence Interval (CI) for the LM scheme is the largest of all CI's obtained with MFCCs as input features in this study, which shows that network configuration obtained with the LM scheme is not advantageous.
- LM algorithm with the choice of the (40-20-10) network is selected to solve the issue of large memory requirement for LM algorithm. And we observed that with the core i5 processor with 2gb of RAM, it is able to run 80 iterations without early termination due to out of memory problem.

- It's also observed that the CG algorithm applied to the (40-20-10) network, converged with about 1000 epochs in about 12 minutes, while the LM algorithm took on average 18 minutes to compute 20-30 epochs. So, CG is preferable over the LM.
- The result also shows that neural network system gives poor output for the speech dataset for which it's not trained.

8.2 Future work

There are still wide variety of the issues that can be addressed in order to build a robust and reliable speech recognizer. Suggestions for future studies include:

Expanding the vocabulary used for speech recognition with new words and increasing the database.

Incorporating speech enhancement techniques to the pre-processing stage to enhance the recognizer accuracy and reliability in the presence of noise.

Investing the performance of other neural network types of word recognition, especially those, which take dynamic nature of the speech into account, such as time-delay neural networks (TDNN), Deep Neural Network (DNN) for example.

References

1. Mohamed, G. Dahl, and G. Hinton (2012), "Acoustic modeling using deep belief networks," IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14–22. 20

2. Besacier, L, Barnard, E, Karpov, A, & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. Speech Communication, 56, 85-100.

3. Black, R. D. (1957), "Ear-insert microphone," Journal of the Acoustical Society of America, Vol. 29, No. 2, pp. 260-264.

4. Boersma, Paul and Weenink, David (2016). Praat: doing phonetics by computer [Computer program]. Version 6.0.15, retrieved 23 March 2016 from http://www.praat.org/

5. Brookes, M. (2005), Voicebox: A Matlab Toolbox for Speech Processing, [http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html], last accessed on November 05, 2005.

6. Chen, K. H, Vu, H. S, Weng, K. Y, Huang, J. H, Tsai, Y. T, Liu, Y. C, &Wang, W. H. (2014). Design of an efficient active noise cancellation circuit for in-ear headphones. In Circuits and Systems (APCCAS), 2014 IEEE Asia Pacific Conference on (pp. 599-602). IEEE.

7. Davis, S. B., and Mermelstein, P. (1980), "Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 4, pp. 357-366.

8. Deller, J. R., Proakis, J. G., and Hansen, J. H. L (1993), Discrete-Time Processing of Speech Signals, Macmillan, New York.

9. Demuth, H, Beale, M, & Hagan (2016), M. Neural network toolbox™ 9.1. User's guide.

10. Duda, R. O., Hart, P. E., and Stork, D. G (2001), Pattern Classification, 2nd Edition, Wiley Interscience, New York.

11. Friesen, L. M, Shannon, R. V, Bas¸kent, D, and Wang, X. (2001), "Speech recognition in noise as a function of the number of spectral channels: Comparison of acoustic hearing and cochlear implants,"J. Acoust. Soc. Am. 110(2), 1150–1163

12. F. Itakura (1975), "Minimum prediction residual applied to speech recognition," IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-23, pp. 67-72, Feb. 1975

13. Gales, M. J. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. Computer speech & language, 12(2), 75-98.

14. Gold, B., and Morgan, N.(2001), Speech and Audio Signal Processing, Wiley, New York.

15. Graciarena, M., Franco, H., Sonmez, K., and Bratt, H.(2003), "Combining standard and throat microphones for robust speech recognition," IEEE Signal Processing Letters, Vol. 10, No. 3, pp. 72-74.

16. H.Sakoe and S.Chiba (1978), Dynamic programming algorithm optimization for spoken word recognition ,IEEE Trans. Acoustics, Speech, Signal Proc., ASSP-26(1).pp.43-49,1978.

17. Hagan, M. T, Demuth, H. B, Beale, M. H, & De Jesús, O (1996), Neural network design (Vol. 20). Boston: PWS publishing company.

18. Harisha, S. B, Amarappa, S, & Sathyanarayana, D. S (2015), Automatic Speech Recognition-A Literature Survey on Indian languages and Ground Work for Isolated Kannada Digit Recognition using MFCC and ANN. International Journal of Electronics and Computer Science Engineering.

19. Hemakumar, G., & Punitha, P. (2013). Speech recognition technology: a survey on Indian languages. International Journal of Information Science and Intelligent System, 2(4), 1-38.

20. Hinton, G. E. (1989). Connectionist learning procedures. Artificial intelligence, 40(1), 185-234.

21. Hopfield, J. J. (1982), "Neural networks and physical systems with emergent collective computational abilities," Proceedings of The National Academy of Sciences, Vol. 79, pp. 2554-2558.

22. Hu YH, Hwang JN (2002) Handbook of neural network signal processing. In/: The electrical engineering and applied signal processing. CRC Press, Boca Raton

23. Huang, L-S., and Yang, C-H.,(2000) "A novel approach to robust speech endpoint detection in car environments," Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, pp. 1751-1754

24. Huang, X. D, Ariki, Y, & Jack (1990), M. A.Hidden Markov models for speech recognition (Vol. 2004). Edinburgh: Edinburgh university press.

25. In Rosenblatt, M (1963), Proceedings of Symposium on Time Series Analysis, Chapter 15, pp. 209-243, Wiley, New York.

26. J.Ferguson,Ed.,(1980),"Hidden Markov models for speech," IDA, Princeton, NJ

27. Junqua, J. C, Mak, B, & Reaves, B (1994), A robust algorithm for word boundary detection in the presence of noise. Speech and Audio Processing, IEEE Transactions on, 2(3), 406-412.

28. Junqua, J-C (1991), "Robustness and cooperative multimodel man-machine communication applications," *Proceedings of Second Venaco Workshop and ESCA ETRW*, pp. 101-112, September 16-20.

29. Kaiser (1990), J. F., "On a simple algorithm to calculate the 'energy' of a signal," *Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 381-384.

30. Krauwer, S. (2003) The basic language resource kit (BLARK) as the first milestone for the language resources roadmap. Proceedings of SPECOM 2003, 8-15.

31. L. Deng and D. Yu (2007) "Use of differential cepstra as acoustic features in hidden trajectory modelling for phonetic recognition," in Proc.ICASSP, pp. 445–448.

32. L.R.Rabiner,(1989),"A tutorial on hidden Markov models and selected applications in speech recognition,"Proc.IEEE,77(2),pp.257-286.

33. Lamel, L. F., Rabiner L. R., Rosenberg, A. E., and Wilpon J. G (1981),"An improved endpoint detector for isolated word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 4, pp. 777-785.

34. Li, Q, Zheng, J, Zhou, Q, & Lee, C. H (2001). Robust, real-time endpoint detector with energy normalization for ASR in adverse environments. In Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP'01). 2001 IEEE International Conference on (Vol. 1, pp. 233-236). IEEE.

35. Lippmann RP (1990), Review of neural networks for speech recognition. In: Readings in speech recognition, pp 374–392. Morgan Kaufmann Publishers, San Mateo.

36. Mitra, S. K (1995), *Digital Signal Processing: A Computer-Based Approach*, 3rd Edition, McGraw-Hill, New York, 2006. Morgan, N., and Bourlard, H. A., "Neural networks for statistical recognition of continuous speech," *Proceedings of The IEEE*, Vol. 83, No. 5, pp. 742-770.

37. Muda, L, Begam, M, & Elamvazuthi, I (2010), Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:1003.4083.

38. Ojanen, E., Ronimus, M., Ahonen, T., Chansa-Kabali, T., February, P., Jere-Folotiya, J., ... & Puhakka, S. (2015). GraphoGame–a catalyst for multi-level promotion of literacy in diverse contexts. Frontiers in psychology.

39. O'Neill, J (1958), "A comparison of mouth, ear, and contact microphones," *The Journal of The Acoustical Society of America*, Vol. 30, No. 7, p. 682.

40. Oppenheim, A. V (1969), "Generalized linear filtering," In Gold, B., and Rader, C. M., editors, *Digital Processing of Signals*, Chapter 8, pp. 233-264, McGraw-Hill, New York.

41. Patel, H. N (2015) Automatic text conversion of continuous speech for indian languages.

42. Patel, J, & Nandurbarkar (2015) A. Development and Implementation of Algorithm for Speaker recognition for Gujarati Language.

43. Picone, J. W (1993), "Signal modeling techniques in speech recognition," *Proceedings of The IEEE*, Vol. 81, No. 9, pp. 1215-1247.

44. Qiang, H., and Youwei, Z. (1998), "On prefiltering and endpoint detection of speech signal,"*Proceedings of The Fourth International Conference on Signal Processing*, Vol. 1, pp. 749-752.

45. R.K.Moore,(1994), "Twenty things we still don't know about speech," Proc.CRIM/ FORWISS Workshop on „Progress and Prospects of speech Research and Technology".

46. Rabiner, L. R.(1989), A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286

47. Rabiner, L. R., and Sambur, M.R.(1975), "An algorithm for determining the endpoints of isolated utterances," *The Bell System Technical Journal*, Vol. 54, pp. 297-315.

48. Rabiner, L. R., and Schafer, R. W. (1978), *Digital Processing of Speech Signals*, Prentice-Hall, New Jersey.

49. Rafaely, B., and Furst, M. (1996), "Audiometric ear canal probe with active ambient noise control," *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 3, pp. 224- 230.

50. Rohini B Shinde and V P Pawar (2012), "A Review on Acoustic Phonetic Approach for Marathi Speech," Recognition. International Journal of Computer Applications 59(2):40-44.

51. Rosenblatt, F. (1958), "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, Vol. 65, pp. 386-408.

52. Rumelhart, D. E., and McClelland, J. L (1986), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol.1, Cambridge, MA: MIT Press.

53. Shahina, A., and Yegnanarayana, B.(2005), "Language identification in noisy environments using throat microphone signals," *Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing*, pp. 400-403, January 4-7.

54. Shen, J. L., Hung, J. W., and Lee, L. S.(1998), "Robust entropy-based endpoint detection for speech recognition in noisy environments," *Proceedings of The International Conference on Spoken Language Processing*.

55. Steinmetz, R. (2012) Multimedia: computing communications & applications. Pearson Education India.

56. Taboada, J., Feijoo, S., Balsa, R., and Hernandez C.(1994), "Explicit estimation of speech boundaries," *IEE Proceedings – Science, Measurement, and Technology*, Vol. 141, No. 3, pp. 153-159.

57. Teager, H. M.(1980), "Some observations on oral air flow during phonation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 5, pp. 599-601.

58. Tailor, J. H, & Shah (2016), D. B., Speech Recognition System Architecture for Gujarati Language. International Journal of Computer Applications, 138(12).

59. UNESCO, E. (2014). Global Monitoring Report Teaching and Learning: Achieving Quality for All.

60. Vaidyanathan, R., Gupta, L., Chung, B., Allen, T. J., Quinn, R. D., Tabib-Azar, M., Zarycki, J., and Levin, J. (2004), "Human-machine interface for tele-robotic operation: mapping of tongue movements based on aural flow monitoring," *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 859-865.

61. Vaidyanathan, R., Kook, H., Gupta, L., and West, J.(2004), "Parametric and non-parametric signal analysis for mapping air flow in the ear-canal to tongue movement: a new strategy for hands-free human-machine interface," *Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, pp. 613-616.

62. Vergin, R., O'Shaughnessy, D., and Farhat, A.(1999), "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *IEEE Transactions on Speech and Audio Processing*, Vol. 7, No. 5, pp. 525-532.

63. Vu, H. S, Chen, K. H, & Fong, T. M (2015), Active noise control for in-ear headphones: Implementation and evaluation. In Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on (pp. 264-265). IEEE

64. Westerlund, N., Dahl, M., and Claesson, I.(2002), "In-ear microphone hybrid speech enhancement," *Proceedings of SIP*, Kauai, Hawai, USA.

65. Westerlund, N., Dahl, M., and Claesson, I.(2002), "Speech recognition in severely disturbed environments combining ear-mic and active noise control," *Proceedings of The 2002 International Congress and Exposition on Noise Control Engineering*,Dearborn, MI.

66. Westerlund, N., Dahl, M., and Claesson, I.(2005), *In-Ear Microphone Techniques for Severe Noise Conditions*, Research Report.

67. Wu, B. F., and Wang, K. C.(2005), "Robust endpoint detection algorithm based on the adaptive band-partitioning spectral entropy in adverse environments," *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 5, pp. 762-775.

68. Ying, G. S, Mitchell, C. D, & Jamieson, L. H (1993), Endpoint detection of isolated utterances based on a modified Teager energy measurement. In Acoustics, Speech, and Signal Processing, 1993. ICASSP-93, 1993 IEEE International Conference on (Vol. 2, pp. 732-735).

69. Ying, G. S., Mitchell, C. D., and Jamieson, L. H. (1993), "Endpoint detection of isolated utterances based on a modified Teager energy measurement," *Proceedings of The IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, pp.732- 735.

70. Zhang, Y., Zhu., X., Hao, Y., and Luo, Y.(1997), "A robust and fast endpoint detection algorithm for isolated word recognition," *Proceedings of The IEEE International Conference on Intelligent Processing Systems*, Vol. 2, pp. 1819-1822.

71. Zhu, Q., and Alwan, A.(2003), "Non-linear feature extraction for robust speech recognition in stationary and non-stationary noise," *Computer Speech and Language*, Vol. 17, No. 4, pp. 381-402.

72. Zue, V. W.(1998) The use of speech knowledge in automatic speech recognition Proceedings of the IEEE, 73(11), 1602-1615.

Publications

Research papers published

1. “Word Boundary Detection for Gujarati Speech Recognition using In-ear Microphone,” India International Conference on Information Processing (IICIP-2016). Paper is published in IEEE Xplore Digital Library.
2. “Neural network based Gujarati Speech Recognition for dataset collected by in ear microphone,” ICACC 2016, 6th International Conference on Advances in Computing & Communications 2016. Paper is published by Elsevier Procedia Computer Science and the publication is made available on sciencedirect.com
3. “Gujarati Speech Recognition System Using Neural Network” International Conference on Soft Computing and Pattern Recognition, SoCPaR 2016. Paper will be published in the Springer proceedings “Advances in Intelligent Systems and Computing”
4. “Recognizing voice commands for robot using MFCC and DTW” International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 5, May 2014
5. “Feature Extraction and Classification Techniques for Speech Recognition: A Review”, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 12, December 2013