People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of

the Requirements for the Degree of

MASTER

In **Telecommunication**

Option: **Telecommunications**

Title:

Speaker Recognition Using Gaussian Mixture Model

Presented by:

Talbi Katia

Supervisor:

Prof.Abdelhakim Dahimene

Registration Number:……/2019

Abstract

Speaker recognition is the process of identifying a person based on its voice characteristics.

A text-independent speaker recognition system using a Gaussian mixture model (GMM) for a set of 10 speakers, selected from TIMIT database, is developed in this project.

The design of an optimum speaker recognition focuses on the selection of parameters that increase the identification rate and minimize false acceptance (FA) and false rejection (FR) errors.

Speaker identification and speaker verification experiments were conducted to evaluate the performance of the system. A variation in the number of GMM components and the dimension of Mel-Frequency Cepstral Coefficients (MFCC) features were studied to select the optimum parameters.

But most importantly the effect of preprocessing the speech signal, using short-time energy and zero crossing rate, at the input of the speaker recognition system have been investigated and compared to the use of a raw speech signal input.

In the training phase an Expectation- maximization (EM) algorithm that is initialized by a K-mean clustering method, was used to estimate the speaker model's parameters.

Finally the speaker recognition decision is based on a maximum likelihood test that is performed in both tasks of speaker identification and speaker verification.

Keywords: GMM, TIMIT, false acceptance (FA), false rejection (FR), MFCC, Identification rate EM algorithm, K-mean.

Dedication

This study is wholeheartedly dedicated to:

My beloved parents who have been my source of inspiration and gave me strength when I needed it the most, who continually provided their moral, spiritual and emotional support

My brothers

My beloved sister in law for her infinite kindness

My dear and lovely nephew

To all my friends and classmates who shared their words of advice and encouragement to finish this study

Acknowledgment

I would like to thank my supervisor Dr Abdelhakim Dahimene,for his patience and guidance throughout this project. He had been one of the most inspiring teachers I had the chance to study with and one who transmitted his tremendous knowledge with great passion and humility. I would also like to thank him for his availability and help whenever I ran into a trouble or had any question about my project.

I take also this opportunity to express my gratitude to the teachers who made my journey at the institute so pleasant, especially Mr. Azrar who encouraged me and supported me and to Miss Cherifi who always strives for the good of her students.

Table of content

Abstract ……i

Dedication ……ii

Acknowledgment ……..iii

Table of content ……iv

List of tables ……vii

List of figures ……viii

List of abbreviations and acronyms ……ix

Chapter 1 1

Introduction 1

1.1. Introduction 1

1.2. Definition of Speaker Recognition 2

1.3. History of Speaker Recognition 3

1.4. Applications of Speaker Recognition 5

1.5. Report Organization 5

Chapter 2 7

Speech Signal 7

2.1. Introduction 8

2.2. Speech Production 8

2.2.1. Speech Production Mechanism 8

2.2.2. Classification of sounds 9

2.2.3. Speech Production Model 10

2.3. Speech Perception 11
2.4. Speech Processing 13
2.4.1. Speech Processing Acquisition 13
2.4.2. Speech Processing Techniques 13
2.4.3. Speech Segmentation 16
2.4.4. Segmentation Algorithm: 18
Chapter 3 20
Feature extraction 20
3.1. Introduction 21
3.2. Feature extraction description 21
3.2.1. Types of features 21
3.2.2. Feature Extraction Methods 22
3.3. MFCC Extraction Procedure 25
3.3.1. Frame blocking 26
3.3.2. Windowing and DFT 27
3.3.3. Mel frequency filter bank: 28
3.3.4. Mel Frequency Cepstrum 29
3.4. Delta and Delta-Delta cepstral coefficients: 30
Chapter 4 32
Speaker Modeling 32
4.1. Introduction 33
4.2. Gaussian Mixture Model (GMM) 33
4.2.1. Motivations 33
4.2.2. Gaussian Mixture Model description: 34
4.3. Expectation-maximization EM 37
4.3.1. Application of EM algorithm in GMM: 38
4.4. Initialization of EM by K-mean clustering 39

4.4.1. Lloyd’s Algorithm (K-means): 40
4.5. Speaker modeling algorithm 41
Chapter 5 43
Experiments and results 43
5.1. Introduction 44
5.2. System Description 44
5.2.1. Database description 44
5.2.2. Implementation Issues 44
5.3. Speaker identification experiment 45
5.3.1. Speaker identification procedure 45
5.3.2. Experiments description: 46
5.3.3. Experiment evaluation 47
5.3.4. Results and discussion 49
5.4. Speaker verification: 54
5.4.1. Speaker verification overview: 54
5.4.2. Speaker verification algorithm: 56
5.4.3. Experiments description and evaluation 57
5.4.4. Results and discussion 57
Chapter 6 *Conclusion* 62
6.1. Conclusion 63
6.2. Suggestion for further research: 63

List of tables

TABLE 1 RESULTS OF THE FIRST SPEAKER IDENTIFICATION EXPERIMENT 49

TABLE 2 RESULTS OF THE SECOND SPEAKER IDENTIFICATION EXPERIMENT 50

TABLE 3 CONFUSION TABLE OF 16 ORDER GMM AND 39 MFCC COEFFICIENTS 51

TABLE 4 RESULTS OF THE THIRD SPEAKER IDENTIFICATION EXPERIMENT 52

TABLE 5 COMPARISON OF THE IDENTIFICATION EXPERIMENTS RESULTS 53

TABLE 6 FIRST SPEAKER VERIFICATION EXPERIMENT 58

TABLE 7 THRESHOLD SELECTION FOR THE FIRST SPEAKER VERIFICATION EXPERIMENT .. 58

TABLE 8 SECOND SPEAKER VERIFICATION EXPERIMENT 59

TABLE 9 THRESHOLD SELECTION FOR THE SECOND SPEAKER VERIFICATION EXPERIMENT 60

TABLE 10 RESULTS COMPARISON OF SPEAKER VERIFICATION EXPERIMENTS 60

List of figures

FIGURE 1.1 INFORMATION CONTAINED IN A SPEECH SIGNAL [3] 1

FIGURE 1.2 SPEAKER RECOGNITION PROCESS, (A) SPEAKER IDENTIFICATION, (B) SPEAKER VERIFICATION [5] 2

FIGURE 1.3 BLOCK DIAGRAM OF TRAINING AND TESTING PHASE OF SPEAKER RECOGNITION [7] 3

FIGURE 2.1 UPPER RESPIRATORY TRACT DIAGRAM 9

FIGURE 2.2 SPECTRUM OF A VOWEL [19] 9

FIGURE 2.3 SPEECH PRODUCTION MODEL [19] 10

FIGURE 2.4 ANATOMY OF THE EAR [21] 12

FIGURE 2.5 PROBABILISTIC PARAMETERS OF HIDDEN MARKOV MODEL (EXAMPLE) [27] 15

FIGURE 2.6 NEURAL NETWORK STRUCTURE 15

FIGURE 2.7 BLOCK DIAGRAM OF VOICED, UNVOICED CLASSIFICATION [30] 19

FIGURE 3.1 BLOCK DIAGRAM OF PLP PROCESSING 23

FIGURE 3.2 MEL SCALE VERSUS LINEAR FREQUENCY 25

FIGURE 3.3 BLOCK DIAGRAM OF MFCC PROCESSOR 26

FIGURE 3.4 FRAMING AN AUDIO SIGNAL TO OVERLAPPING FRAMES [42] 27

FIGURE 3.5 MEL FILTERBANK WITH 10 FILTERS 28

FIGURE 4.1 UNIVARIATE GAUSSIAN MIXTURE MODEL WITH TWO COMPONENTS 34

FIGURE 4.2 GAUSSIAN MIXTURE MODEL PARAMETERS 36

List of abbreviations and acronyms

ASR	Automatic Speaker Recognition/ Automatic Speech Recognition
DCT	Discrete Cosine Transform
DFT	Fast Fourier Transform
DTW	Dynamic Time Warping
EM	Expectation Maximization
FA	False Acceptance
FFT	Discrete Fourier Transform
FR	False Rejection
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
Idr	Identification Rate
LPC	Linear Prediction Coding
MFCC	Mel Frequency Cepstral Coefficients
ML	Maximum Likelihood
PDF	Probability Density Function
STE	Short Time Energy
VAD	Voice Activity Detection
VQ	Vector Quantization
ZCR	Zero Crossing Rate

Chapter 1

Introduction

1.1. Introduction

Biometrics refers to the automatic identification of a living person based on physiological or behavioral characteristics [1]. Biometric identification is preferred over traditional identification methods that involve password and pins, because they are more intuitive to the user which makes them more convenient.

Various types of biometric systems are being used for real time identification; the most popular are based on face recognition and fingerprint matching. Furthermore, there are other biometric systems that utilize iris and retinal scan, face, hand geometry and voiceprint. [1]

Voiceprint is one of the most unique forms of identification that a person can produce, it's far more complex than any other biometric component since it contains a combination of information, **Figure 1.1**, like a person's accent, inflexion and rhythm as well as physical factors related to the size and shape of person's vocal tract [2].

The process of authentication using a voiceprint is known as voiceprint recognition or more commonly used speaker recognition.

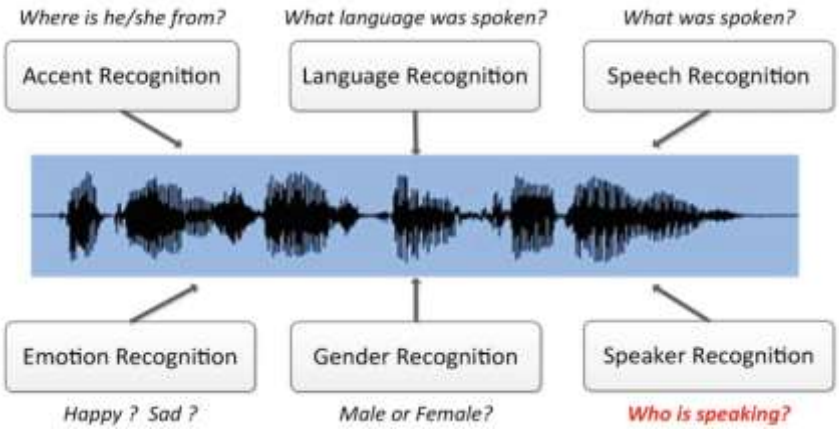

Figure 1.1 Information contained in a speech signal [3]

1.2. **Definition of Speaker Recognition**

Speaker recognition is the process of automatically recognizing a speaker's identity based on specific information carried through its produced speech waveform.

We distinguish two main tasks in speaker recognition, speaker identification and speaker verification, **Figure 1.2** .Speaker identification aims to identify an input speech by matching it to one model from a set of known speaker models, whereas speaker verification aims to identify whether an input speech corresponds to a claimed identity which is considered as biometric authentication where the voice is used as password [4]

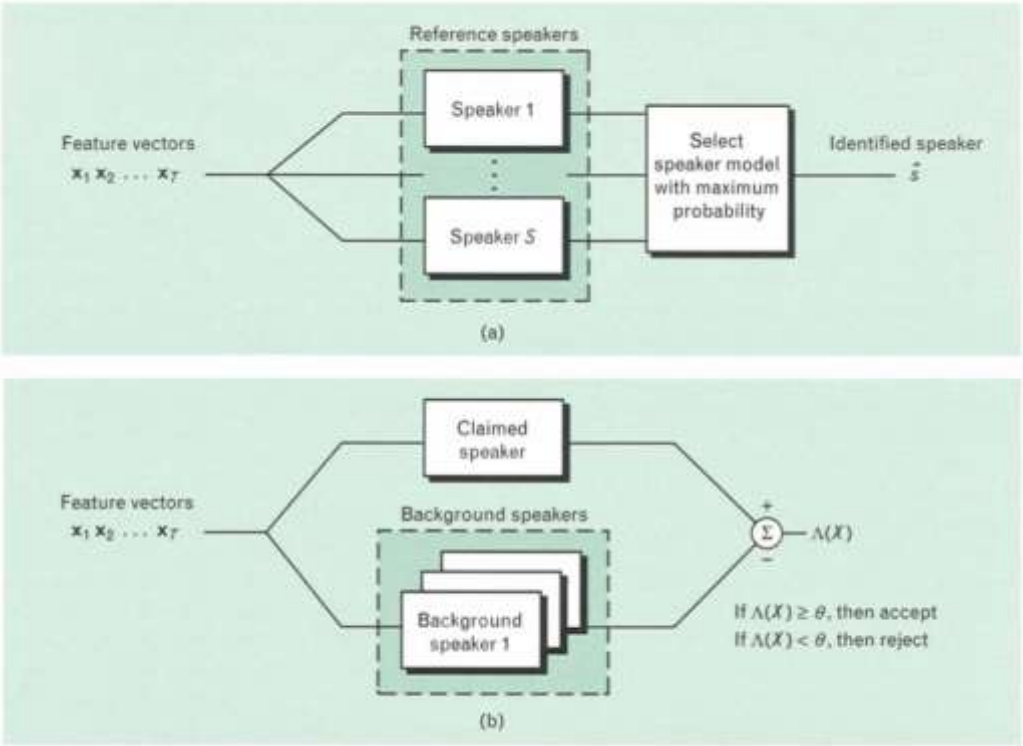

Figure 1.2 speaker recognition process, (a) speaker identification, (b) speaker verification [5]

The speaker recognition system passes through enrollment and verification phase. The enrollment phase consist of recording a speaker's voice and extracting important information that characterize it to form a model, **Figure 1.3**. During verification a speech sample (utterance) is compared against the created models in the

enrollment phase such that in identification task the comparison is made against multiple models to find the best match ,whereas the verification task include a comparison against only one model . [6]

Speaker recognition systems can be classified to two categories including text dependent and text independent recognition. Text dependent speaker recognition implies that a speaker must utter specific key words or sentences having the same text in enrollment and verification phase. Text independent recognition in the other hand can identify the speaker regardless of what is being said.

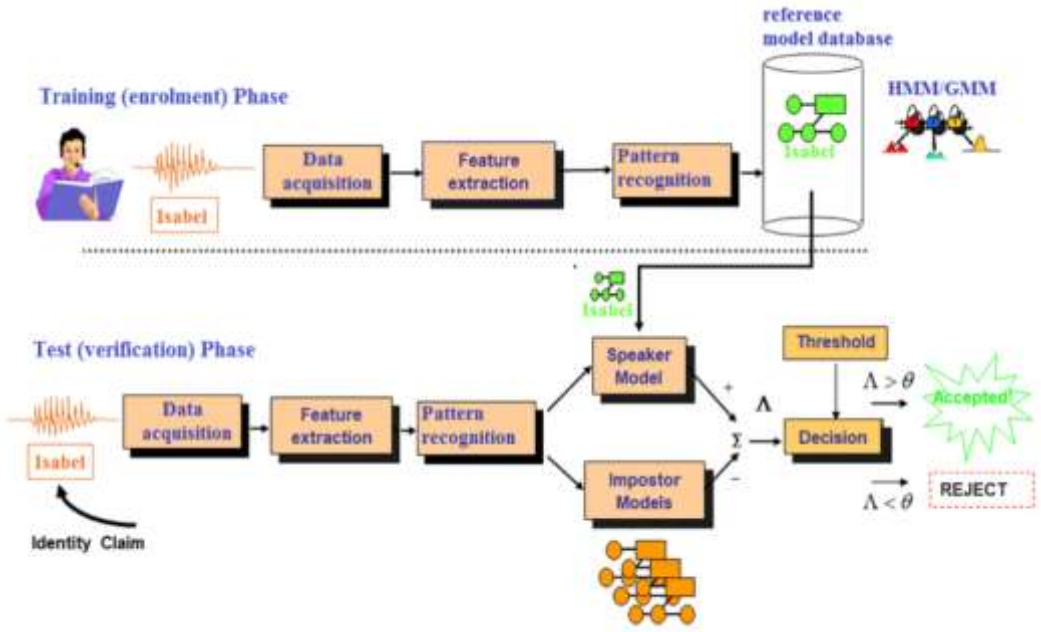

Figure 1.3 Block diagram of training and testing phase of speaker recognition [7]

1.3. **History of Speaker Recognition**

Research in automatic speech and speaker recognition has now spanned five decades and their progress can is summarized as follows [8]:

- **1960s and 1970s:**

1960 announced the first attempt for automatic speaker recognition after a being preceded by a decade of speech recognition research. Pruzansky at Bell Labs [9] was among the first to initiate research by using filter banks and correlating two

digital spectrograms for a similarity measure, then it has been developed by Doddintgon [10] at Texas Instruments (TI) replacing filter banks by formant analysis and TI built the first fully automated large scale speaker verification system providing high operational security. Endres et al. [11] and Furui [12] had also investigated on intra speaker variability a major problem in speaker recognition systems.

- **1980s**

During this decade the Hidden Markov Model (HMM) method, [13]was introduced as an alternative to the template-matching approach used previously for text-dependent speaker recognition and it remarkably increased its performance. This technique used speaker models derived from a multi-word sentence, a single word, or a phoneme.

For text independent recognition nonparametric and parametric models were investigated during this decade .As a non-parametric model Vector Quantization (VQ) [14] was introduced it was based on the compression of a short time training vectors to a small set of points called VQ codebook.As parametric model Pritz proposed to use ergodic HMM while Rose et al. [15] proposed to use a single state HMM which is now called Gaussian mixture model (GMM) and is considered as state of the art in text independent speaker recognition.

- **1990s:**

Research on increasing robustness became a central theme in the 90's and different methods have been investigated like the Text-prompted method proposed by Matsui et al. [16] Where the key sentences are changed each time the system is used meaning that the system accepts the input utterance only when it determines that the registered speaker uttered the prompted sentence. The score normalization has also been investigated to normalize the variation of likelihood that result from the same speaker (intra-speaker variation) using Likelihood ratio- and a posteriori probability-based techniques.

- **2000s:**

New normalization techniques have been proposed in which the scores are normalized by subtracting the mean and dividing over the standard deviation that have been estimated from the imposter score distribution. High level features have also been successfully used in text independent speaker verification, basically this method takes high level features like word idiolect, pronunciation or prosody and produces a sequence of symbols from the acoustic signal to perform recognition using the frequency and the co-occurrence of symbols. [8]

1.4. **Applications of Speaker Recognition**

The general area of speaker recognition is authentication, surveillance and forensics. The authentication is mostly used in security application, like credit cards transaction where speaker authentication is combined with other biometric techniques to reinforce the security of transactions. Speaker recognition is also used as a mean of surveillance in security agencies to find relevant information about target speakers of interest for the service. But the most important application of speaker recognition is the forensic which is very helpful when a speech sample is recorded during a crime, so that a suspect's voice can be compared to detect similarities between the two voices [17]

1.5. Report Organization

This first chapter introduced the basic concepts of speaker recognition and its evolution through history, showing that it's a field that is still evolving and still need some improvement especially the text -independent recognition and since Gaussian mixture model is a powerful method for that task, this thesis will go through the mains steps to develop a text-independent speaker recognition system using Gaussian mixture model (GMM).

For that purpose, the 2nd chapter will discuss about speech signal, how it is produced, how it is perceived and how it processed by computers. The goal is to gain an understanding of the speech in general.

Chapter 3 will present the idea of feature extraction and show some of the common used method, then the extraction of Mel-frequency cepstrum coefficients will be developed in more details to be further used for the project. Chapter 4 will explain the concept of speaker modeling using Gaussian mixture model (GMM) and how different of its parameters are estimated by the EM algorithm that is initialized by a Kmean classifier. The algorithm and the initialization procedure will also be encountered in this chapter

Chapter 5 will describe the experimental sets up utilized, like the database and the different trainings and testings that were performed before each experiment, the results of each experiment will be summarized in a table and discussed, and finally chapter 6 will contain the conclusion of the thesis and further research that can improve the recognition system.

Chapter 2

Speech Signal

2.1. Introduction

To plan a speech-based interface system, it is important to comprehend the working of the human auditory system [18]. At the linguistic level of communication, an idea is first formed in the mind of the speaker, and then produced in the form of speech. The idea is transformed into words, phrases, and sentences according to the grammatical rules of the language. Finally, when the idea reaches the listener's linguistic level, the brain performs speech recognition and understanding.

This chapter will encounter the basics of speech signal, how it is produced, how it is perceived and how it is proceed by computers.

2.2. Speech Production

2.2.1. Speech Production Mechanism

The speech signal represent a sequence of sounds .Theses sounds and the transition between them carry the information that need to be conveyed. [19]

The sequence of sounds follows certain rules, Linguistics is the study of such rules, whereas the study of the classification of the basic sounds is called phonetics.

In order to come up with a model of speech production, we need to have an understanding of the human vocal system. It consists of two main parts: the vocal cords (or glottis), and the vocal tract see **Figure 2.1**. The vocal tract in turn consists of three main parts:

- The pharynx – connection from the esophagus to the mouth.
- The oral cavity – the mouth.
- The nasal tract – begins at the velum and ends at the nostril

The source of energy comes from the air pressure exerted by the lungs, bronchi and trachea. Speech is produced when an acoustic wave is radiated from this vocal system when air is expelled from the lungs and the air flow is perturbed by constrictions somewhere in the vocal tract. When the velum is lowered, the nasal tract is acoustically coupled to the vocal tract to produce nasal sounds.

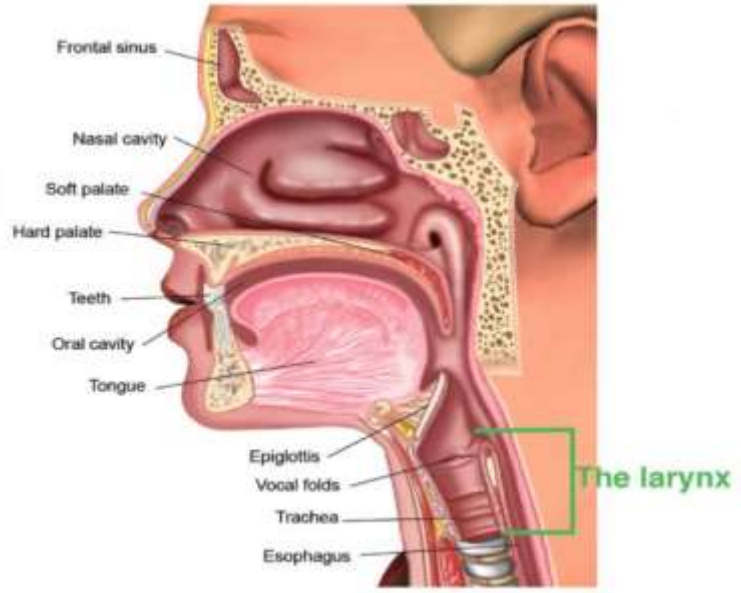

Figure 2.1 Upper respiratory tract diagram

2.2.2. Classification of sounds

The basic units of speech sounds in the English language are called phonemes. There are two main types of phonemes: vowels and consonants. More detailed classifications are also available. But we shall assume only these two types of sounds. Vowels are produced when the vocal tract is excited by pulses of air caused by the vibration of the vocal cords. The vibration is periodic in nature and the period is the pitch of that sound. The shape of the vocal tract determines the resonant frequencies of the tract, called formants. For vowels, there are typically three formants between the frequencies 200 Hz and 3 kHz. The exact frequencies of the formants vary from person to person. **Figure 2.2** shows a typical frequency spectrum of a vowel

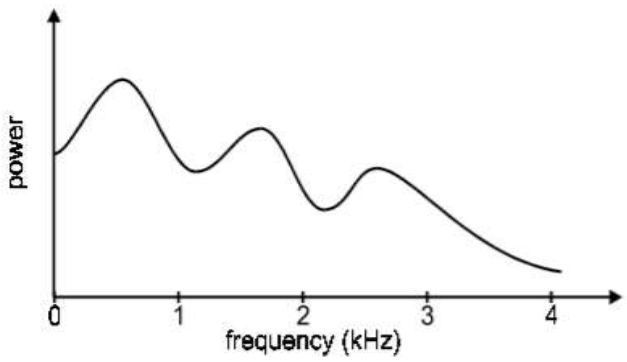

Figure 2.2 Spectrum of a vowel [19]

In the production of consonants, the vocal cord is totally relaxed in general, although there are exceptions. In this way, air flows into the vocal tract without the periodic excitation generated by the vocal cord. Consonants can be broadly classified into:

• **Nasals:** are produced when the vocal tract is totally constricted at some point along the oral cavity. The velum is lowered and the air flows through the nasal tract, radiating through the nostrils.

• **Fricatives:** are produced when the steady air flow becomes turbulent in the region of a constriction in the vocal tract.

• **Stops:** are transient sounds produced by building up pressure behind a total constriction somewhere in the oral tract, and suddenly releasing the pressure.

2.2.3. Speech Production Model

In order to synthesize speech sounds artificially; we need a model of the speech production system described above. Figure 2.3 shows a more detailed model.

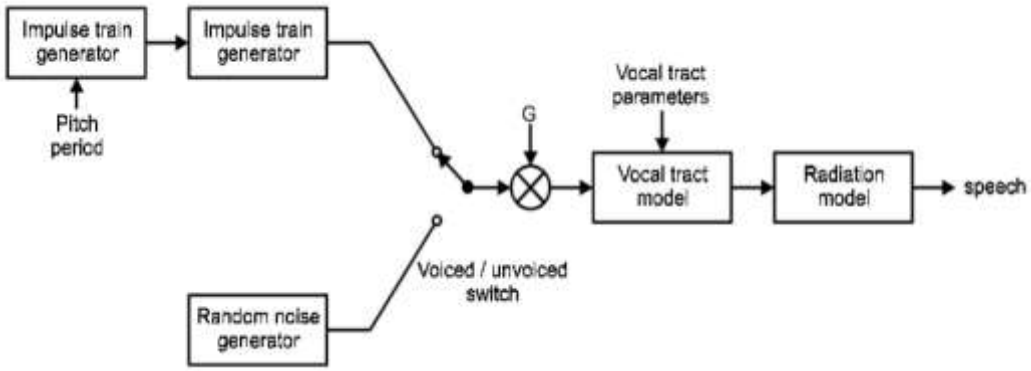

Figure 2.3 Speech Production Model [19]

The glottal pulse model, the vocal tract model, and the radiation model are linear discrete-time systems. They are therefore essentially discrete-time filters. In order to synthesize speech, the voiced/unvoiced switch will switch to the source for the sound at that particular time. The vocal tract parameters will also need to vary with time. One of the most successful glottal pulse models is the Rosenberg model. Its impulse response is given by:

$$g(n)=\begin{cases}\frac{1}{2}\left[1-\cos\left(\frac{\pi N}{N_1}\right)\right] & \text{for } 0 \le n \le N_1 \\ \cos\left[\frac{\pi(n-N_1)}{2N_2}\right] & \text{for } N_1 \le n \le N_1+N_2 \\ 0 & \text{otherwise}\end{cases} \quad (2.1)$$

The vocal tract model is usually a linear predictive model. It is called so because the current speech sample is generated from a number of past samples plus the current excitation. This can be described in equation form:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + u(n) \quad (2.2)$$

Such that:

a_k: is the coefficient for the model and it changes from one phoneme to another.

u (n): is the input sample to the vocal tract model.

p: is the prediction order and typically ranges from 10 to 12. [19]

2.3. Speech Perception

Hearing is the well-known process that allow us to perceive the vibrations that can cause a sound, however, perception is not just a mode of hearing, rather it is how the sound is interpreted and made sense of. This implies that the same sound could be perceived differently by two listeners. [20]

The process of perceiving a sound begins at the level of the sound signal that reaches the listener's ear and trigger the process of audition.

The ear is related to the brain by three main parts: the outer ear, the middle ear and the inner ear, see **Figure 2.4** .Each part plays a role in extracting the information carried by the heard sound.

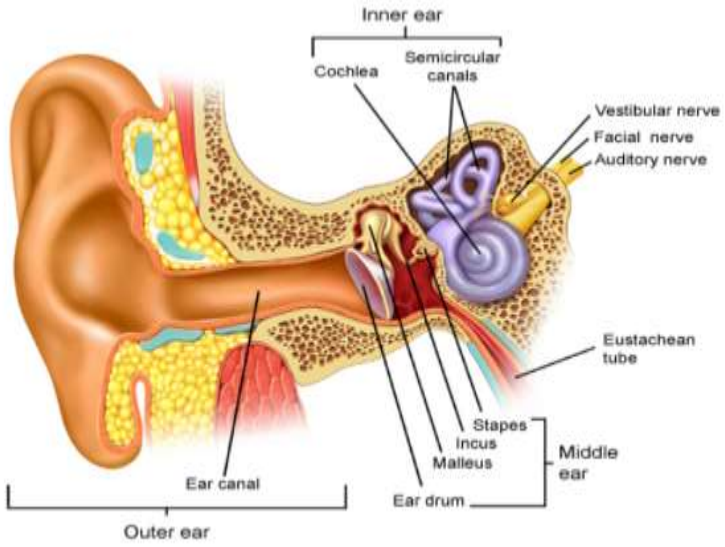

Figure 2.4 Anatomy of the ear [21]

The outer ear includes the pinna and the ear canal such that the perceived sound travels through the pinna down to the canal ear, the pinna is also responsible of detecting the direction from which the sound came from.

At the end of the ear canal the eardrum is found and this announce the start of the middle ear. This part include three bones called ossicles and they form a chain from the eardrum to the inner ear. When the sound hits the eardrum it makes it move back and forth and depending on the nature of the sound and its pitch the movement will differ and will make the small bones move producing a signal that will be received by the inner ear.

The inner ear helps both with the hearing and the balance .The cochlea is the hearing part whereas the semicircular canals helps us keep our balance. [21]

The cochlea has a bony structure and looks like a snail, it comprises a fluid and hair cells .When the middle ear bones move, the fluid of the inner ear moves to and causes the movement of some hair cells, since not all hair cells are receptive to the same type of sound.

The hair cells transform the movement into electrical signals that passes through the auditory nerve to reach the brain that will process it to understand the meaning of the sound and how it should respond.

2.4. Speech Processing

Speech processing is the study of speech signals and processing methods. The signals are usually processed in a digital representation, so speech processing can be regarded as an intersection of digital signal processing and natural language processing, applied to speech signals. Aspects of speech processing include the acquisition, manipulation, storage, transfer, and output of speech signals [18, 22]

Speech processing technologies are used for digital speech coding, spoken language dialog systems, text-to-speech synthesis, and automatic speech recognition. Information (such as speaker gender, or language identification, or speech recognition) can also be extracted from speech. [18]

2.4.1. Speech Processing Acquisition

Speech processing extracts the desired information from a speech signal and process it by a digital computer, thus the signal must be represented in digital form so that it can be used by the computer. Devices such as microphones and telephone handsets can be used to convert a received acoustic wave to an analog signal .The obtained signal is conditioned with antialiasing filtering that limits the bandwidth of the signal to approximately the Nyquist rate that represent half of the sampling rate. The analog signal is sampled and passed to the analog to digital (A/D) converter. [23]

Today's A/D converters for speech applications typically sample with 12–16 bits of resolution at 8000–20000 samples per second. Oversampling is commonly used to allow a simpler analog antialiasing filter and to control the fidelity of the sampled signal precisely. In local speaker-verification applications, the analog channel is simply the microphone, its cable, and analog signal conditioning. Thus, the resulting digital signal can be very high quality, lacking distortions produced by transmission of analog signals over long-distance telephone lines. [23]

2.4.2. Speech Processing Techniques

Dynamic Time warping and hidden Markov model are the most widely used techniques for speech processing and their brief description is presented below:

2.4.2.1. Dynamic Time Warping (DTW)

During the last decade this technique became widely used in speech processing [22]. DTW is an algorithm that measures similarities between two temporal sequences which may vary in time or speed. [24]

Dynamic Time Warping (DTW) was originally designed to treat automatic speech recognition such that when a word is recorded and needs to be matched to another one, the two signals appear to be very similar ,however their length and their features look different, thus to measure the similarity of the two signals DTW algorithm is performed [24] .

In general DTW is a method that calculates an optimal match between two given sequences. It focuses on matching two sequences of feature vectors by repetitively shrinking or expanding the time axis following certain restrictions until an exact match is obtained [25, 26].

2.4.2.2. Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are a class of probabilistic graphical model that allow us to predict a sequence of unknown (hidden) variables or states from a set of observed variables, see **Figure 2.5.** It can be viewed as a Bayes Network unrolled through time with observations made at a sequence of time steps being used to predict the best sequence of hidden states. [27]

The application of HMM in speech recognition assumes that the hidden variables are the produced phonemes whereas the observed data are the small frames of audio signal that are represented by feature vectors. So given a set of feature vectors HMM is used to predict the produced sequence of phonemes that are interpreted to words using phoneme to word dictionary. [28]

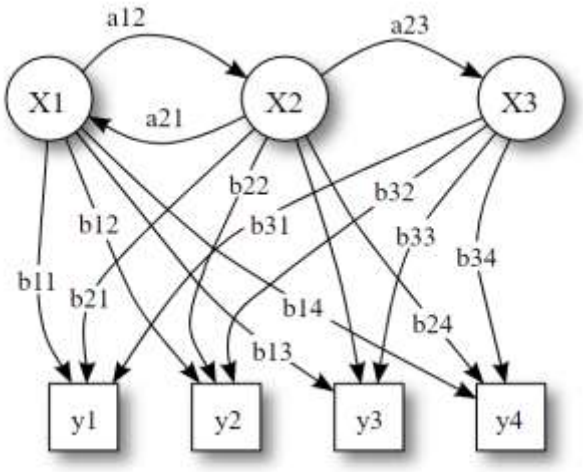

Figure 2.5 Probabilistic parameters of hidden Markov model (example) [27]

2.4.2.3. Neural Network

The Generalization is the strength of artificial neural network. It provides a processing to simulate information that is analogues to the human nervous system. Multilayer feed forward network with back propagation algorithm is commonly used in classification and pattern recognition which makes it suitable in speaker recognition applications.

The neural network is structured into input layer, hidden layer and output layer, **Figure 2.6.** Such that each layer is composed of a certain number of neurons.

The number of input neurons is chosen to be the same as the total number of features whereas the output layer has the same number of neurons as the speakers that need to be recognized, however this number need to be set in the hidden layer after performing multiple tests by varying the number of neurons, then choose the one that gives the best results.

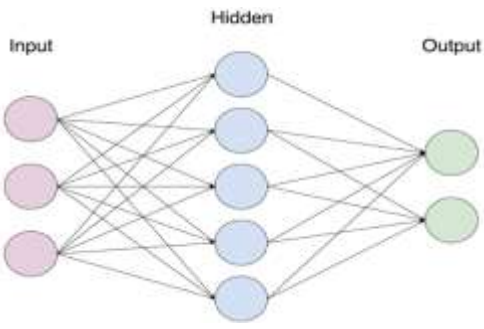

Figure 2.6 Neural network structure

2.4.3. Speech Segmentation

Many speech processing systems require segmentation of speech signal into its basic acoustic units, thus, segmentation can be defined as the process of breaking down a speech signal into smaller units and it represent a primary step in any voiced activated system like speech recognition. [29]

Before segmenting a speech into smaller units, an understanding of some of its characteristics is needed

2.4.3.1. Speech Signal characteristics

A continuous speech signal comprises two parts, the part that carries the speech information and the second part that carries silence and noise, holding useless information. The informative section of speech can further be classified to voiced and unvoiced speech. The Voiced sound is generated when the air flows through the larynx and the vocal cords are semi closed, see **Figure 2.1**, whereas unvoiced sound are produced when the vocal cords are open. [29]

Voiced speech signal is approximated by a slow changing periodic signal with a frequency caused by the vibration of the vocal cords and which is different from one speaker to another. This frequency is known as the pitch and usually male's pitch ranges from 50Hz to 250Hz while the female's contribute with a pitch between 120Hz and 500Hz.This suggests that the energy of voiced speech is concentrated at low frequencies , below about 3 kHz. [30, 29]

Unvoiced speech signals in the other hand does not exhibit any periodic components and appears to have some similarities with a noisy signal. Most of their energy is concentrated at high frequencies [29, 30]

Therefore a speech signal can be considered as a sequence of voiced and unvoiced sounds that are smoothly connected, in addition to that silence regions represent an integral part of speech that determines the separation between different utterances and which is considered as a background noise.

2.4.3.2. Types of Features in Speech Segmentation

Different features can be extracted from a speech signal to help us in its segmentation, we distinguish mainly time domain features and frequency domain

features. Two time domain features techniques will be further presented, the short time signal energy and the zero crossing rate, two techniques that are easy to implement and which can be combined to perform an efficient segmentation that separate a signal to voiced/unvoiced parts.

- **Short-Time Signal Energy**

Short time energy is the simplest feature that can be extracted from a speech signal. Speech signal energy is computed on a short-time basically by windowing the signal at a particular time, squaring the obtained quantity and computing their average. [29]

The square root of the result is an engineering quantity known as the Root Mean Square (RMS).

The short-time energy function of a speech frame with length N is defined as:

$$E_n = \frac{1}{N}\sum_{m=-\infty}^{\infty}[x(n-m)w(m)]^2 \quad (2.3)$$

The short-term root mean square (RMS) energy of this frame is given by:

$$E_{n(Rms)} = \sqrt{\frac{1}{N}\sum_{m=-\infty}^{\infty}[x(n-m)w(m)]^2} \quad (2.4)$$

Such that: x (n) is the discrete-time audio signal and w (n) is rectangle window function of length N:

$$w(n) = \begin{cases} 1, & 0 < n < N-1 \\ 0, & otherwise \end{cases} \quad (2.5)$$

In general the amplitude of unvoiced speech segment is much lower than the amplitude of voiced segment and the energy of speech signal is representative of such amplitude variations making this method efficient at detecting the voiced sounds in a speech signal which tend to have higher short time energy compared to the unvoiced and silence segments.

- **Short-Time Average Zero-Crossing Rate**

The average zero-crossing rate refers to the number of times speech samples change algebraic sign in a given frame [29]. The rate at which zero crossings occur

is a simple measure of the frequency content of a signal , Since high frequencies imply high zero crossing rates, and low frequencies imply low zero-crossing rates. [30]It measures the number of times in a given time interval or frame the amplitude of the signal passes through a value of zero and it is represented by the following equation:

$$Z_n = \frac{1}{2}\sum_{m=-\infty}^{\infty}\left[\text{sgn}[x(n-m)] - \text{sgn}[x(n-m-1)]\right]w(m) \quad (2.6)$$

Such that: sign is the signum function defined in equation (2.7) and w (m) is the rectangular window, equation (2.5).

$$\text{Sgn}[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) < 0 \end{cases} \quad (2.7)$$

Since there is a correlation between the zero crossing rate and the energy distribution with frequency we can conclude that unvoiced segments of a speech signal are characterized by a higher zero crossing rate than its voiced parts [30].

2.4.4. Segmentation Algorithm:

Each time domain feature presented above can be used to segment the speech into voiced, unvoiced and silent parts therefore by combining short time energy with zero crossing rate we can develop a segmentation algorithm .

As mentioned previously short time energy classify high energy frames as voiced segments and the remaining parts are unvoiced .The zero crossing rate in the other hand considers that high ZCR in a segment indicate an unvoiced part of the speech. Therefore segments that record high energy and low zero crossing rate will be set as voiced segments of speech.

The proposed algorithm is going to be used as a preprocessing step in the experimental part of the project. It is summarized in **Figure 2.7** and explained below:

- **Step1**: we divide the signal into N non overlapping frames , with 256 samples in each frame
- **Step2** :calculate the energy E and zero crossing rate ZCR for each frame
- **Step3**: compare the obtained results with energy threshold (E_{th}) and zero crossing rate threshold (ZCR_{th})
- **Step4:** Verify the condition :

For i=1,…, N

$$\text{if } \begin{cases} E(i) > E_{th} \\ \text{and} \\ ZCR(i) < ZCR_{th} \end{cases} \quad \text{Then save the frame i as a voiced segment}$$

Else repeat from step1 by dividing frame i to two frames of 128 samples

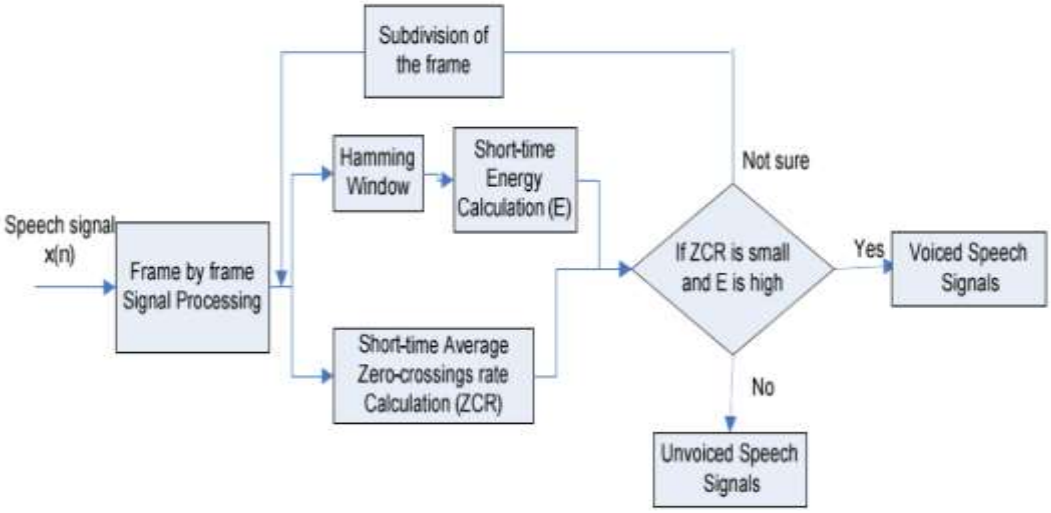

Figure 2.7 Block diagram of voiced, unvoiced classification [30]

Chapter 3

Feature extraction

3.1. Introduction

An automatic speaker recognition system (ASR) is a biometric system that needs to identify the distinctive attributes of each speaker, these attributes are referred to as features and the process of identifying them is known as feature extraction.

Features are informations extracted from the speech waveform and are represented in the form of numerical data. For an effective recognition algorithm the features must be informative, discriminative and independent [31].

Different techniques of feature extraction are used, some of them will be presented briefly in this chapter, however the most prevalent and dominant technique for speech and speaker recognition applications is the MFCC that have been chosen for this project and will be presented in more details through the upcoming sections.

3.2. Feature extraction description

The main goal of this step is the computation of a sequence of feature vectors which provide a compact representation of the speech signal.

3.2.1. Types of features

A variety of features have been developed for speaker recognition purpose and they can be divided to the following classes: [32]

- Spectral features
- Dynamic features
- Source features
- High-level features

Spectral features describe the short-term speech spectrum that hold more or less information about the physical characteristics of the vocal tract .Dynamic features reflect the time evolution of the spectral features .Source features captures the glottal voice source characteristics. Finally the high-level features describe features that have a symbolic type of information like characteristic word usage.

In our application only spectral features will be considered and are explained below:

- **Spectral feature extraction**

The performance of feature extraction goes through three main stages: [33]

1) The first stage is speech analysis or the acoustic front-end, it performs spectra-temporal analysis of the speech signal and generates raw features describing the envelope of the power spectrum of short speech intervals.
2) The second stage extend the feature vector by combining static and dynamic features.
3) The last stage transforms the extended feature vectors into more compact and robust vectors that are then supplied to the recognizer.

3.2.2. Feature Extraction Methods

The most widely used features for speaker recognition are described below:

3.2.2.1. Linear Predictive Coding (LPC):

LPC is a technique used for low or medium rate coder that are usually found when transmitting a speech signal through a wireless media and where it is desirable to compress the signal for efficient storage and transmission. It is one of the most powerful speech analysis techniques and it has gained popularity as a formant estimation technique, i.e. concentration of acoustic energy around a particular frequency in the speech wave.

When the speech signal is passed through a filter that remove the redundant bits it generate the residual error that need to be suppressed and obviously this error is quantized by a smaller number of bits than the original signal. So instead of transmitting the entire signal we can transmit the residual error and some speech parameters to be able to reconstruct the original signal at the destination. These parameters constitute a parametric model that is computed based on the least mean squared error theory known as the linear prediction (LP) method. [33]

The LPC analyses the speech signal by estimating the formants, then remove them to estimate the intensity and frequency of the residue. Such that it will synthesize the original speech signal by reversing the process using the residue parameters to create a source signal and the formants to create a filter [34]. Both combined yield the original signal.

The main idea of LPC method is to predict the value of the current sample by a linear combination of previous already reconstructed samples, [35] see **equation (3.1)** And then to quantize the difference between the actual value and the predicted value **equation (3.2)** .

$$\widehat{s_n} = -\sum_{k=1}^{p} a_k s_{n-k} \quad (3.1)$$

Such that p is the order of LPC analysis and $\{a_1, a_2, \ldots, a_p\}$ are the LPC coefficients

The error between the actual value and the predicted one can be computed as

$$e_n = s_n - \widehat{s_n} \quad (3.2)$$

Or: $$e_n = s_n + \sum_{k=1}^{p} a_k s_{n-k}$$

Since $\{e_n\}$ is obtained by subtracting $\{\widehat{s_n}\}$ from $\{s_n\}$, it is called the residual signal.

3.2.2.2. Perceptual Linear Prediction (PLC)

The Perceptual Linear Prediction PLP model developed by Hermansky in 1990 [37], models the human speech based on the concept of psychophysics of hearing. PLP discards irrelevant information of the speech and thus improves speech recognition rate. PLP is similar to LPC except that its spectral characteristics have been transformed to match characteristics of human auditory system. [34]

PLP approximates three main perceptual aspects namely: the critical-band resolution curves, the equal-loudness curve, and the intensity-loudness power-law relation, which are known as the cubic-root. Shown in **Figure 3.1**

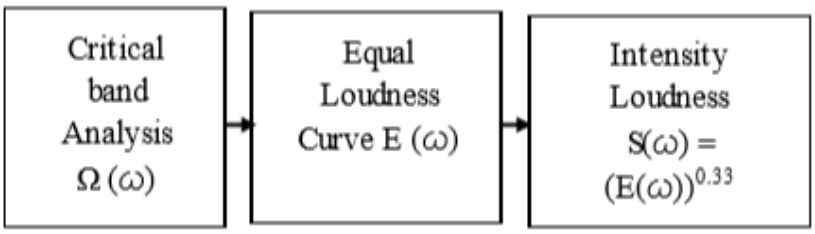

Figure 3.1 Block diagram of PLP Processing [34]

To better represent the human hearing resolution, PLP method wraps the spectrum of the speech signal into the Bark scale. To find the bark frequency corresponding to an audio frequency **equation (3.3)** is used.

$$(\omega) = 6\ln\left[\frac{\omega}{1200\pi} + \left[\left(\frac{\omega}{1200\pi}\right)^2 + 1\right]^{0.5}\right] \quad (3.3)$$

3.2.2.3. Mel Frequency Cepstral Coefficients (MFCC)

Mel frequency cepstral coefficients (MFCCs) are features widely used in ASR, they were first introduced by Davis and Mermelstein in the 1980's [36]**,** and have been state-of-the-art ever since.

To know more about the MFCC's we need to expand our notions concerning the Mel scale and the cepstral domain.

The Mel scale is a perceptual scale that was named by Stevens, Volkmann and Newman in 1937 [37].It's a scale that relates the perceived frequency of a tone to the actual measured frequency [38] based on a led experiments on human subjects. This experiment showed that we are much better at discerning small frequency changes at low frequencies which are smaller than 1 kHz than at higher frequencies, therefore the Mel scale is represented linearly below 1 KHz and follows a logarithmic scale for higher frequencies as shown in **Figure 3.2**.

We can illustrate this concept by taking a tone at 300Hz and another one at 400Hz, we will notice that our brain can detect that the distance between the two is small, however if we hear a tone at 900Hz and another one at 1KHz we perceive a higher distance than in the first case even if they are actually the same [39]. The Mel scale was developed to capture such differences and including this scale makes the extracted features match more closely what humans hear

The Mel scale is related to the linear frequency scale by the equation:

$$M=2596log_{10}\left(1+\frac{f}{700}\right) \quad (3.4)$$

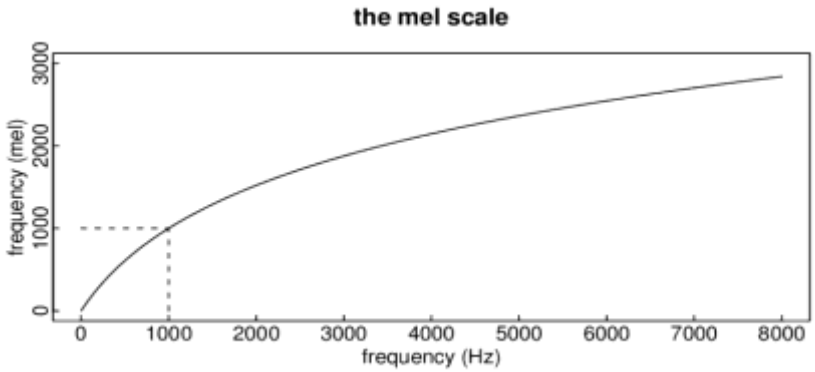

Figure 3.2 Mel Scale versus linear frequency

In the other hand, the word cepstral in the acronym of MFCC is the word spectral with 'spec' reversed and the reason behind this is that when doing Fourier transform on a time signal we obtain a spectrum that is its representation in the frequency domain , however when we take the magnitude of this spectrum and apply a cosine transformation on its logarithm , the resulting spectrum is neither in frequency nor in time domain , hence bogert et al [40] decided to call this domain **quefrency** , and the spectrum of the log of the spectrum of the signal in time domain is what is called cepstrum [39] .

3.3. MFCC Extraction Procedure

To compute the MFCC parameters six major steps need to be followed and they are illustrated in **Figure 3.3**. The first one consist of dividing the speech signal into small frames such that the speech waveform appears to be stationary with respect to time and we choose frames of 25ms with overlapping between two adjacent frames of about 50%(±10%) then to minimize the discontinuities caused by the framing we apply a hamming window at each frame, the third step consist of applying a discrete Fourier transform (DFT) for the frames and take their magnitude. We apply to this result a Mel frequency filter bank which transform the signal from frequency to the Mel scale and then we take its logarithm to finally apply a discrete cosine transform (DCT) which result in Mel frequency cepstrum coefficients.

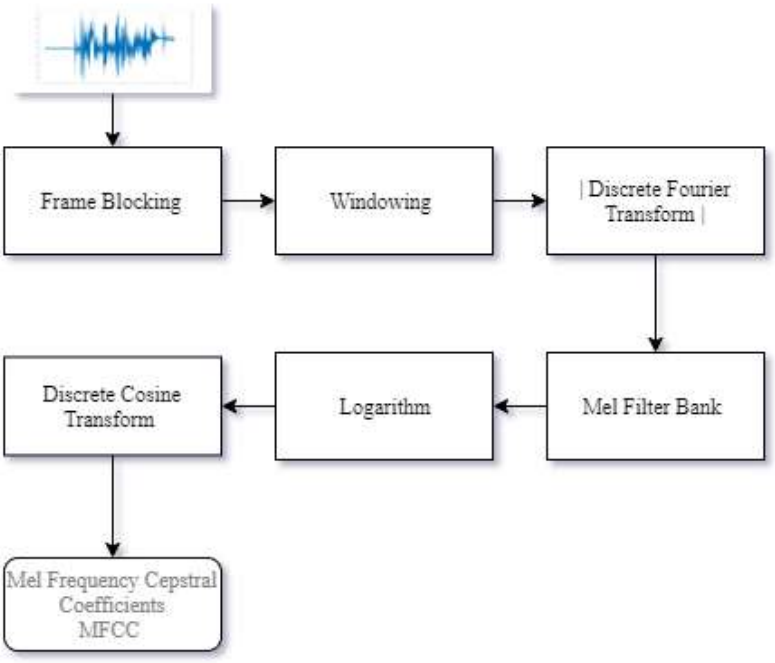

Figure 3.3 Block Diagram of MFCC Processor

3.3.1. Frame blocking

Let x[n] be a continuous speech signal sampled at a frequency f_s .

x[n] is a signal that varies constantly with respect to time so for simplicity we assume that on short time scales of 20ms to 40ms x[n] is stationary and divide it into P frames, each having a duration of 25ms that corresponds to N samples with an overlap of N/2 samples shown in figure3 .Now we can represent x[n] in matrix notation as

$$\chi=[\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3},\ldots, \overrightarrow{x_l},\ldots, \overrightarrow{x_P}]$$

Such that $\overrightarrow{x_l}$ represent the l^{th} frame of of the speech signal x[n] of size Nx1 yields the matrix χ with dimension N x P [41].

Note that MFCC coefficients are computed for each frame.

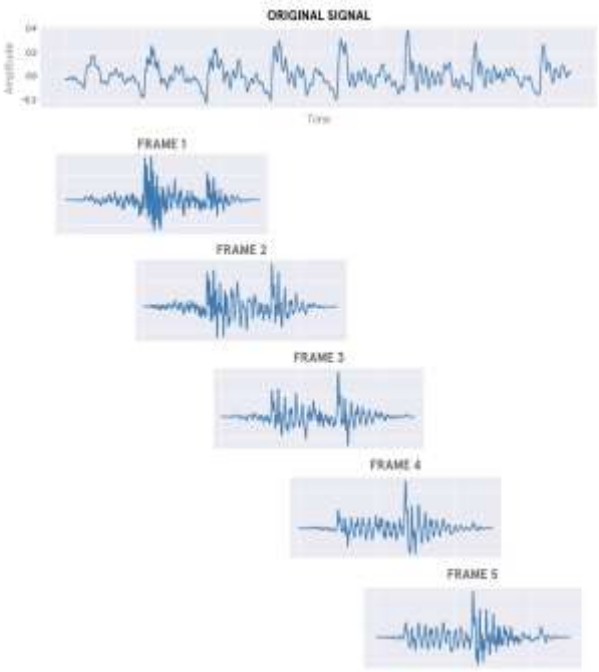

Figure 3.4 Framing an audio signal to overlapping frames [42]

3.3.2. Windowing and DFT

To minimize the signal discontinuities we apply a hamming window to each frame.

The hamming window equation is given by:

$$W[n] = 0.54 - 0.46 \cos\left(\frac{n\,\pi}{N}\right) \tag{3.5}$$

Taking the l^{th} frame $\overrightarrow{x_l}$ we multiply it by the hamming window followed by a discrete Fourier transform DFT yields:

$$X_l(k) = \sum_{n=0}^{n=N-1} x_l[n]w[n]\, e^{\frac{-j2\pi kn}{N}} \tag{3.6}$$

Such that k=0…N-1, so a DFT is computed for each sample in the frame l, and the corresponding frequency of the k^{th} sample is $f(k) = k\frac{f_s}{N}$.

Computing the DFT points of the N samples in frame l result in a vector $\overrightarrow{X_l}$ of sizeN:

$$\overrightarrow{X_l} = [X_l(0), X_l(1), X_l(2), \ldots X_l(N-1)]^T$$

This represent the DFT of the windowed l^{th} frame $\overrightarrow{x_l}$ of the speech signal x[n], however, these operations are performed on each of the P frames and we obtain a vector of size N for each one, so the DFT of the matrix χ is a matrix of size NxP

$$X= [\overrightarrow{X_1}, \overrightarrow{X_2}, \overrightarrow{X_3},\ldots, \overrightarrow{X_l} \ldots, \overrightarrow{X_p}]$$

The operation of dividing a time varying signal into equal frames and computing their Fourier transform separately to determine its overall Fourier transform is known as short time Fourier transform or STFT and in our study X is an STFT matrix that is complex with phase and magnitude but we are only concerned about the information carried in its magnitude which is extracted from the modulus of the DFT and represented by |X| [41]

3.3.3. Mel frequency filter bank:

The magnitude spectrum |X| is a matrix of size N x P that is represented on a linear frequency scale. In this step we will wrap this spectrum according to the Mel scale and to do so we need to divide |X| into critical bands using the Mel filter bank that consist of a series of overlapping band pass filters that have a triangular shape(see **Figure 3.5**) and are defined by :

- The central frequency f_c
- The number of filters used F
- The minimum frequency f_{min}
- The maximum frequency f_{max}

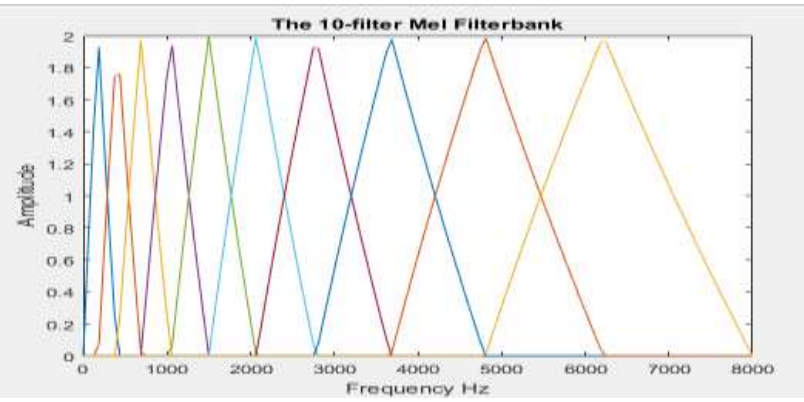

Figure 3.5 Mel filterbank with 10 filters

At first we compute the frequency resolution in the Mel scale using this equation:

$$\delta\phi_f = \frac{\phi_{f_{max}} - \phi_{f_{min}}}{F+1} \quad (3.7)$$

Where $\phi_{f_{min}}$ and $\phi_{f_{max}}$ are the Mel frequencies corresponding to the linear frequencies f_{min} and f_{max} respectively.

Using the frequency resolution in the Mel scale we can compute the center frequencies of the triangular filters in the mel scale using equation:

$$\phi_{fc}(m) = m.\delta\phi_f \quad (3.8)$$

Such that m=1…F

The next step is to find the triangular center frequencies in the linear scale using the inverse of equation 1 which is:

$$f_c = 700(10^{\phi_{fc}(m)/2595} - 1) \quad (3.9)$$

Now that we have the central frequencies we can write the equation of the Mel filter bank

$$M(m,k) = \begin{cases} 0 & \text{for } f(k) < f(m-1) \\ \frac{f(k)-f_c(m-1)}{f_c(m)-f_c(m-1)} & \text{for } f_c(m-1) \le f(k) < f_c(m) \\ \frac{f(k)-f_c(m+1)}{f_c(m)-f_c(m+1)} & \text{for } f_c(m) \le f(k) < f_c(m+1) \\ 0 & \text{for } f(k) \ge f(m+1) \end{cases} \quad (3.10)$$

The mel filter bank is an F x N matrix

3.3.4. Mel Frequency Cepstrum

The output of the filter bank corresponds to the product of the magnitude spectrum |X| with the Mel filter bank M (m,k) .The logarithm operation on this output result in the Mel cepstrum $L_P(m,k)$ of the speech signal x[n] such that :

$$L_l(m,k) = \ln\{\sum_{k=0}^{k=N-1} M(m,k) * |X_l(k)|\} \quad (3.11)$$

Where m =1, 2,. . ., F and l=1, 2, . . . ,P

So for one frame we obtain a mel cepstrum value for each of the F filters and since we have P frames , $L_l(m,k)$ is a matrix of dimension F x P .

Finally the MFCC parameters are computed by a discrete cosine transform DCT of $L_l(m,k)$ using this equation:

$$\phi_l^r\{x[n]\}=\sum_{m=1}^{F} L_l(m,k)\cos\{\frac{r(2m-1)\pi}{2F}\} \quad (3.12)$$

Such that r= 1, 2,…,F where $\phi_l^r\{x[n]\}$ represent the r^{th} MFCC of the l^{th} frame

The MFCC's of all the P frames of the speech signal x[n] is obtained as:

$$\phi\{\chi\}=[\ \phi_1, \phi_2, \ldots, \phi_l, \ldots, \phi_P\ ] \quad (3.13)$$

Each column in the matrix $\phi\{\chi\}$ corresponds to the MFCC coefficients of one frame of the speech signal[n] so it has a dimension of Fx P .

The MFCC data sets represent cepstral acoustic vectors and they are used as feature vectors in our project, however, it is possible to obtain more details about speech features using a derivation on the MFCC. This approach permits the computation of the delta MFCC (DMFCCs), as the first order derivatives of the MFCC. Then, the delta-delta MFCC (DDMFCCs) are derived from DMFCC, being the second order derivatives of MFCCs. [43]

3.4. Delta and Delta-Delta cepstral coefficients:

Delta and delta-delta cepstras are also known as differential and acceleration coefficients respectively, they are evaluated based on MFCCs.

The motivation behind adding these two features to the already existing MFCCs is that it seems that this later captures only the power spectral envelope of a single frame, nevertheless it has been shown that speech contains also some informations in the dynamics (the trajectories of the MFCC coefficients over time) And it turns out that computing the time derivatives of the standard static MFCC that have been found previously increase the speaker recognition performances. [38]

The delta cesptral coefficient is calculated using the formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} c_{i+\theta} - c_{i-\theta}}{\sum_{\theta=1}^{\Theta} \theta^2} \quad (3.14)$$

Where d_t is the delta coefficient at time t, computed in terms of the corresponding static coefficients $c_{i-\theta}$ to $c_{i+\theta}$ and Θ is the size of delta window. [43]

The delta-delta coefficients are computed with the same formula but are applied to the delta coefficients rather than the MFCCs

Chapter 4

Speaker Modeling

4.1. Introduction

The previously extracted features will be used to create a model for each speaker. The model is essential in capturing the gradual changes of a speaker's voice, since it is unrealistic to ask the user to utter all the possible utterances across different sessions, especially for text-independent based authentication.
The solution is to build speaker models using a small amount of data and whenever an utterance of an unknown speaker is inputted to the system, it will be compared with each speaker's model and the closest match will determine the identity of the speaker.

Depending on the application of the speaker recognition system, different modeling techniques are used, but broadly they are classified into generative and discriminative models.

The generative models use training data samples from a target speaker and estimate its feature distribution to form a statistical or a non-parametric model, it includes models like Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), and Vector Quantization (VQ).

However the discriminative models use training data of target and non-target speakers to learn how to optimally separate between each speaker and model the boundaries between them, it includes models like Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). [44]

In this project we choose to use a statistical generative model which is the Gaussian Mixture Model (GMM) that outperforms the other generative models in text-independent speaker recognition application like it has been demonstrated by Reynolds and Rose experiment [15]

4.2. Gaussian Mixture Model (GMM)

4.2.1. Motivations

The Gaussian mixture model is a statistical model used to represent a normally distributed subpopulations within an overall population, such that the different subpopulations are unknown and can be learned automatically. [45]

There is two main motivations behind choosing this model to represent a speaker's identity.

The first one is that a speaker's voice is characterized by a set of acoustic classes where each class can be seen as a representation of a phonetic event like vowels, nasals, or fricative. These classes hold information about the vocal tract configuration of the speaker, which is useful to its identification and each class has a spectral shape that can be represented by a mean μ and a covariance matrix Σ, however, these classes are unknown and need to be learned to be able to classify the feature vectors into phonetic events, therefore it has been shown that by assuming independent feature vectors, their density drawn from this hidden classes is considered as a Gaussian mixture. [15]

The second motivation is the ability of GMM to form smooth approximations for arbitrarily-shaped densities (features) using a discrete set of Gaussian functions, each with its own mean and covariance matrix which allow a better modeling capability. In some sense it can be seen as hybrid between the unimodal Gaussian that represent a speaker's feature by position (mean vector) and an elliptic shape(covariance matrix) and the VQ model that represent speaker's distribution by a discrete set of characteristic templates. [15]

4.2.2. Gaussian Mixture Model description:

A Gaussian mixture model is parameterized by three types of values, the mixture component weight, the mean and the variance.

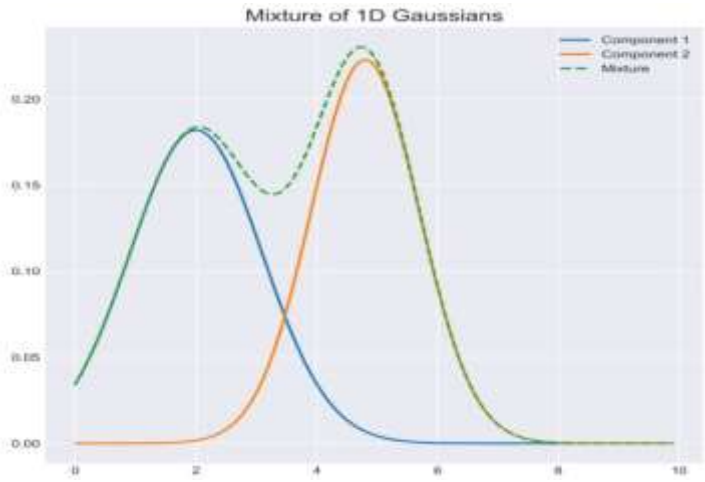

Figure 4.1 Univariate Gaussian mixture model with two components

For a Gaussian mixture model with M mixtures the k^{th} component has a mean μ_k and a variance $\boldsymbol{\sigma}_k$ for the univariate case, **Figure 4.1**. However, multivariate mixtures are represented by a mean vector $\overrightarrow{\mu_k}$ and a covariance matrix Σ_k .The mixture weight for the k^{th} component is represented by a value p_k with the constraint that $\sum_{i=0}^{k} p_i$ =1 such that it evaluate the probability that a data point x was generated by the k^{th} component of the Gaussian mixture and when it is learned it constitute the a-posteriori probability of the component given the data.

The univariate Gaussian mixture model density:

$$P(x)=\sum_{i=0}^{M} p_i \, N(x|\mu_i, \sigma_i) \tag{4.1}$$

Such that:

$$N(x|\mu_i, \sigma_i) = \frac{1}{2\sigma_i\sqrt{2\pi}} \exp\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right) \tag{4.2}$$

And:
$$\sum_{i=0}^{M} p_i = 1 \tag{4.3}$$

D-Multivariate Gaussian mixture model density:

$$P(\vec{x}) = \sum_{i=0}^{M} p_i N(\vec{x}|\overrightarrow{\mu_i}, \Sigma_i) \tag{4.4}$$

Such that:

$$N(\vec{x}|\overrightarrow{\mu_i}, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_i|}} \exp\left(\frac{-1}{2}(\vec{x}-\overrightarrow{\mu_i})^T\Sigma_i^{-1}(\vec{x}-\overrightarrow{\mu_i})\right) \tag{4.5}$$

And:
$$\sum_{i=0}^{M} p_i = 1$$

Since our data (features) is multidimensional we will be concerned only with the multivariate case, therefore the complete Gaussian mixture density is parameterized by a mean vector a covariance matrix and mixture weights, **Figure 4.2**, which are represented by the notation:

$$\lambda = \{p_i, \overrightarrow{\mu_i}, \Sigma_i\} \qquad i=1,2,\ldots,M$$

For speaker identification purpose, each speaker is represented by its model λ, and depending on the type of parameters selected for the model, different results are obtained.

We can choose to use a nodal covariance which means a covariance matrix for each component of the GMM, or a grand covariance, one covariance matrix for all the Gaussian components. We can also choose a full or diagonal covariance. [15]

In our project the selected covariance is nodal and full.

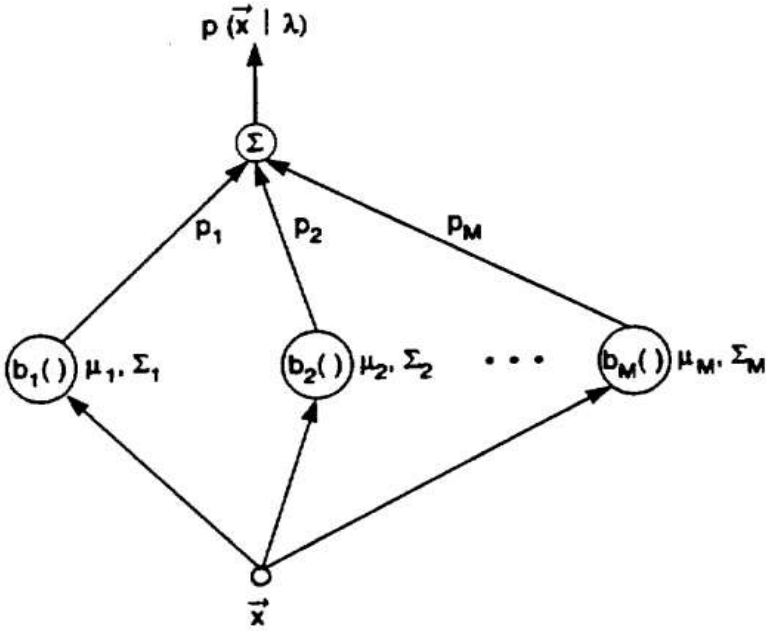

Figure 4.2 Gaussian mixture model parameters

To obtain an optimum model for each speaker a good estimation of the GMM parameters need to be computed and the most popular technique is the maximum likelihood estimation (ML).

Maximum likelihood is used on a training set to estimate the parameters that maximizes the probability (likelihood) of the observed data given the model parameters, see **equation (4.6).**

$$P(X|\lambda)=\prod_{i=0}^{T} P(\vec{x_i}|\lambda) \quad (4.6)$$

Such that $\{\vec{x_1}, \vec{x_2},\ldots,\vec{x_T}\}$ is a sequence of T training vectors. [46]

Unfortunately, this is a nonlinear function of the parameters λ and finding the maximum likelihood estimation solution analytically is not possible.

The solution is to estimate the ML parameters iteratively using a special case of expectation- maximization algorithm (EM).

4.3. Expectation-maximization EM

The Expectation maximization algorithm is a method used to find maximum likelihood estimate of parameters having a certain distribution when the data is incomplete or has a missing value. [47]

An application of the EM algorithm generally begins with the observation that the likelihood function L (λ; X) can be optimized if a set of additional variables is known, this set is referred as missing '.

By assuming that the observable data X is 'incomplete data' and by adding the missing variables Z we obtain a 'complete data', Y ,the probability that links the missing variables to the actual data is written as P(y,z|x,λ) .

Applying the logarithm to the density of P yields the 'complete-data likelihood' Lc (λ; Y) (which is a random variable due to Z), whereas the original likelihood L (λ; X) is the 'incomplete-data likelihood'.

The first 'E'step of the EM algorithm aims to estimate the expected value of the 'complete-data likelihood' giving the observed data X and the **current** model λ as:

$$Q(\lambda; \lambda^{(i)}) = E(L_c(\theta; Y)|X) \tag{4.7}$$

Such that $\lambda^{(i)}$ represent the estimated parameters at the i^{th} iteration and are computed with respect to the parameter $\lambda^{(i-1)}$

Q is a deterministic function that is maximized through the second 'M'step of the EM algorithm, to find the new parameters λ^{i+1} such that:

$$\lambda^{(i+1)} = \arg\max_{\lambda} Q(\lambda; \lambda^{(i+1)}) \tag{4.8}$$

The aim of this process is to improve the complete likelihood, therefore, from an iteration to the next we retain the parameters that increase the value of Q, however our interest is to improve the likelihood of our given data which has been referred to as the 'incomplete likelihood' [48]. The relationship between this two likelihoods has been demonstrated by **Dempster, et al**. to be proportional and an increase in Q implies an increase for the likelihood of the given data, meaning that at each iteration the obtained model is improved:

$$L(\lambda^{(i+1)}; X) \geq L(\lambda; X)$$

Equality is reached only for stationary point of L, making the likelihood increase monotonically and in practice this means convergence to a local maximum. [48]

4.3.1. Application of EM algorithm in GMM:

In our project, the observed data are the extracted features but this is considered as an incomplete data, since they are generated by acoustic classes which are unknown.

Briefly the application of the EM algorithm to a GMM consist at first, to estimate to which mixture (class) C_k, each training vector($\vec{x}_i \in X$) belongs , using an initial model $\lambda^{(0)}$, then, this expectation is maximized and a new model λ is obtained .

More precisely the two steps consist of:

4.3.1.1. The expectation step (E)

This step uses initial parameters p_0, $\overrightarrow{\mu_0}$, Σ_0 to compute the expectation component assignments C_k of each $\vec{x}_i$.

For every mixture component C_k the following computation is performed:

For ∀ i, k:

$$\widehat{\gamma_{ik}} = P(C_k | \overrightarrow{x_i}, \lambda^{(0)}) = \frac{p_k N(\overrightarrow{x_i} | \widehat{\mu_k}, \widehat{\sigma_k})}{\sum_{j=1}^{k} p_i N(\overrightarrow{x_i} | \widehat{\mu_j}, \widehat{\sigma_j})} \quad (4.9)$$

Such that $\widehat{\gamma_{ik}}$ is the a posteriori probability for the acoustic class C_k.

4.3.1.2. The maximization step (M)

The purpose here is to maximize the expectations computed in the first step and this yields new parameters $\overrightarrow{p_1}$, $\overrightarrow{\mu_1}$ and Σ_1 that constitute a new model λ^1

For each mixture component C_k and each training vector $\vec{x}_i$ we have:

Mixture weight:

$$\widehat{p_k} = \frac{1}{T} \sum_{i=1}^{T} \widehat{\gamma_{ik}} \quad (4.10)$$

Mean vector:

$$\overrightarrow{\hat{\mu}_k}=\frac{\sum_{i=1}^{T}\widehat{\gamma_{ik}}\ \overrightarrow{x_i}}{\sum_{i=1}^{T}\widehat{\gamma_{ik}}} \tag{4.11}$$

Variance

$$\widehat{\sigma_k^2}=\frac{\sum_{i=1}^{T}\widehat{\gamma_{ik}}(\overrightarrow{x_i}-\hat{\mu}_k)^2}{\sum_{i=1}^{T}\widehat{\gamma_{ik}}} \tag{4.12}$$

The new model is used as an initial model for the first step in the next iteration and the process is repeated until convergence is achieved.

From an algorithmic perspective, the dominant practical method for estimating GMMs is the Expectation-Maximization (EM) algorithm [49] , but this method is only guaranteed to converge to a stationary point of the likelihood function which can be local instead of a global maxima.

To guarantee convergence to a near globally optimal solution. It is required for the EM algorithm be provided with a reasonable initialization parameters.

There are many different clustering algorithms that can be used in initialization, but so far the most popular algorithm is the K-mean, that is also known to be a suitable candidate for the starting configuration of the EM algorithm.

4.4. Initialization of EM by K-mean clustering

The basic idea behind K-mean clustering is to partition the data into clusters, such that all the data points that have similar attributes are grouped into the same cluster whereas dissimilar data are classified in different groups.

More precisely K-mean clustering assumes that each cluster in the data can be represented by a cluster center, and the data from a cluster will be closer to their cluster centers .Based on this assumption, the goal of K-mean clustering is to find the cluster labels l_i that minimize the 'within cluster sum of squares' (WCSS). [50]

Given a set of observation $\{x_1, x_2,\ldots,x_T\}$. K-mean partition the T observations into K sets $S= \{s_1,s_2,\ldots,s_K\}$:

$$\text{WCSS}=\sum_{k=1}^{K}\sum_{i=1}^{T} z_{ik}\|x_i-\mu_k\|^2 \tag{4.13}$$

Such that: $\mu_k=\frac{\sum_{i=1}^{T} z_{ik}x_i}{\sum_{i=1}^{T} z_{ik}}$ and $\|\,.\,\|$ denotes the Euclidian distance.

The solution to this equation is hard to find, even when K=2, therefore an approximate algorithm is used and the standard one for the K-mean clustering is the Lloyd's algorithm which is also known as the K-mean algorithm. It is an iterative algorithm that update the data cluster's assignments at each iteration, until WCSS stops improving [50].

4.4.1. Lloyd's Algorithm (K-means):

Input:

- Data X
- Number of clusters K
- Initial cluster centers (centroids): $\{\mu_1^{[0]}, \mu_2^{[0]},\ldots, \mu_K^{[0]}\}$

Condition: While WCSS increases at each iteration n **do**:

Cluster Assignment:

Assign each data point to its closest cluster center:

$$z_{ik}^{[n]}=1 \text{ if } \left\|x_i-\mu_k^{[n]}\right\|=\min_{1\le j\le K}\left\|x_i-\mu_j^{[n]}\right\|$$

Update cluster centers:

The cluster centers are updated based on the assignments:

$$\mu_k^{[n+1]}=\frac{\sum_{i=1}^{T} z_{ik}^{[n]}x_i}{\sum_{i=1}^{T} z_{ik}^{[n]}} \tag{4.14}$$

Output:

- Cluster indicator vectors for each data point: $\overrightarrow{z_1}, \overrightarrow{z_2},\ldots, \overrightarrow{z_T}$

The cluster indicator vectors will then be used as an initialization for the EM algorithm.

Some of the motivations behind using this algorithm is that it's guaranteed to converge to a local minimum of WCSS, it runs very efficiently and it does not require a

lot of input parameters, Moreover, the K-mean clustering is considered as a special case of GMM that has a spherical covariance matrix that is the same for every component, which makes sense to choose K-mean algorithm as an initialization configuration for an EM algorithm fitting a GMM. [50]

4.5. Speaker modeling algorithm

Now we can use our theoretical background to set an algorithm that create a model for each speaker.

As described previously the optimal model is obtained by first using the k-means algorithm that will then serve as an initialization parameters to the EM algorithm which finally will estimate the Gaussian mixture model's parameters.

The upcoming steps will describe the algorithm used for this task :

1. **Collect the data :**
   - Extract the MFCC feature vectors from all the spoken utterances of one speaker and choose the ones to use as training and testing.
   - The training feature vectors $\{\overrightarrow{x_1}, \overrightarrow{x_2},\ldots,\overrightarrow{x_T}\}$will be concatenated and used as input data

In this project 10 speakers from TIMIT data base [see chapter5] have been selected, 8 of their utterances were used for training and 2 for the testing.

2. **Choose the desired sets up for training :**
   - Number of the mixtures
   - Number of dimensions
   - Nodal and full covariance matrix

The number of mixtures that give the best performance is hard to choose, that's why we train our model with three different number of mixtures [16, 20, 24] and see how it affects the performance of the system.

The number of dimensions determine the number of the feature vectors. By adding the delta and delta-delta coefficients to the 13 MFCC's we end up with 39 dimensions in total.

3. **Set a function for the training :**

A loop is created to run the following process 20 times and compute the negative log-likelihood in each iteration to finally take GMM parameters that obtained the smallest value of negative log likelihood which is equivalent to the maximization of the likelihood.

- Run the K-mean algorithm on the input data and save the results
- Set up the maximum iteration number to 500 for the EM trainer
- Perform the EM algorithm on the output of the K-mean

The statistics and machine learning toolbox in matlab provide a function that fit a GMM using EM algorithm and compute in each operation the negative log likelihood which need to be saved.

4. **Choose the best model :**
   - compare the negative log likelihood of all the iterations
   - save the GMM parameters λ_1 that minimize the negative log likelihood
5. **Train other speakers :**
   - The same process is repeated for each of the 10 selected speakers to find λ_i for i=1, 2,…, 10.
6. **Final output**
   - Save all the obtained model $\{\lambda_1, \lambda_2,\ldots, \lambda_{10}\}$ to be used further in the identification and verification experiments .

This chapter explained the basics of speaker modeling and its importance in speaker recognition application, different methods were introduced and we decided to use GMM's in our project.

A brief description about GMM's showed how its parameters are used to create speaker models λ_i and how EM algorithm is used to ease their estimation, we also showed the tendency of this algorithm to converge to local maximum and how we can avoid this using good initialization parameters computed through the K-means clustering algorithm.

Following these steps a speaker modeling algorithm was presented to train a GMM and to choose the best output parameters.

Chapter 5

Experiments and results

5.1. Introduction

This chapter will cover the performed speaker identification and speaker verification experiments, the used sets up will be explained and, the obtained [49] results will be developed for each experiment.

The identification and verification systems will be described and the performance of the proposed algorithm in the previous chapter will be evaluated.

5.2. System Description

5.2.1. Database description

The TIMIT database will be used for all the upcoming experiences.

TIMIT corpus of read speech was designed by Texas Instruments, Inc. and MIT to provide speech data for acoustic phonetic studies and for the development of automatic voice-based recognition systems. [51]

TIMIT database contains speech recordings of 630 speakers from eight American regions representing the eight major dialects of American English, each speaker has 10 recording sentences, each with a duration that varies from 2 to 5 seconds .The first 2 sentences named SA are read by all the speakers and are meant to expose their dialectal variants. The 5 other sentences named SX are phonetically-compact sentences and were designed to give a good coverage of pairs of phones and each seven speakers read the same phonetically compact sentences, Finally the last 3 sentences are phonetically-diverse sentences named SI, chosen from existing text sources to add diversity in sentence type and phonetic context .Each of the 3 sentences were different and read by a single speaker.

Each sentence is saved in a speech waveform file with a sample frequency of 16 kHz.

5.2.2. Implementation Issues

5.2.2.1. Model order:

The choice of the number of Gaussian mixture components is not easy to make, because it has a major effect on the performance of the speaker identification system such

that when only few components are used the model can't cover all the phonetic events that are produced by the speaker and an under-fitted model is created.

However, choosing a high order GMM will result in an over-fitted model which means that the present noise in the training data will be taken into consideration and will be represented as a phonetic event of the speaker.

5.2.2.2. Dimension of MFCC features:

The other factor that plays a major role in speaker identification performance is the number of MFCC features that are used. Knowing that the lower order coefficients contain most of the information about the vocal cords and vocal tract shape of the speaker, choosing 13 cepstral coefficients (including the o th coefficient that represent the average power of the input signal) seems to be a good choice, however taking into consideration the time variation of these coefficients which are the delta and delta-delta coefficients is proved to give better results, [38] with the constraint of increasing the feature's dimension from 13 to 39 resulting in more complex models that need higher training duration.

5.3. Speaker identification experiment

This section will develop the followed procedure in the evaluation of a text-independent speaker identification system that uses MFCC features and GMM.
The experimental phase will contain three main parts that will evaluate the identification system by varying different parameters like GMM order and MFCC dimension to optimize the identification rate.

5.3.1. Speaker identification procedure

The first step in any speaker identification experiment is to choose a set S of N speakers S= {1, 2,… ,N} such that each speaker has a set of speech recordings represented by samples at a sampling frequency fs.

For a speech signal of duration t a set of feature vectors $\{\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots\ldots, \overrightarrow{x_t}\}$ is extracted

The third step consist of creating a GMM model for each speaker using the extracted features, this results in a set of N models $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$

The final step in speaker identification, also called the testing phase is to identify the speaker's identity $\hat{s}$ whose model has the maximum posterior probability from an input feature-vector sequence.

The minimum-error Bayes rule for this problem is:

$$\hat{s}=\arg \max_{1\le s\le S} \Pr(\lambda_s|X)= \arg \max_{1\le s\le S} \frac{P(X|\lambda_s)}{P(X)} \Pr(\lambda_s) \quad (5.1)$$

Assuming equal prior probabilities of speakers $\Pr(\lambda_s) = 1/S$ and $P(X)$ are constant for all speakers then they can be ignored in the maximum. Using logarithms and assuming independence between observations, the decision rule for the speaker identity becomes [5]:

$$\hat{s}=\arg \max_{1\le s\le S} \sum_{t=1}^{T} \log p(x_t|\lambda_s) \quad (5.2)$$

Such that $p(x_t|\lambda_s)$ is given in equation (4.6)

5.3.2. Experiments description:

First experiment: 10 speakers from TIMIT database were chosen randomly such that one male and one female were selected from five different regions. 8 sentences from each speaker were used as a training data comprising an average duration of 20 seconds, whereas the testing phase use 2 utterances from each speaker with a total duration of 6 seconds.

The MFCC features are extracted directly from the raw speech signal without any segmentation .The identification rate is measured for different values of MFCC coefficients (13 and 39) and for three varying number of Gaussian mixture components (16, 20, and 24).

Second experiment: The 10 speakers of TIMIT that have the longest utterances were selected for this experiment in order to increase the duration of the training data from an average of 20 seconds to 35 seconds. The set of speakers is constituted of 3 females and 7 males randomly spread over 6 different regions. The identification rate is evaluated by varying the number of MFCC coefficients and mixture components as in the first experiment.

Third experiment: The dataset of the 2nd experiment is used and a preprocessing of data is performed before the feature extraction phase such that the speech was segmented to silent, unvoiced and voiced parts. Only the voiced parts were used as training and testing data. The identification rate is measured following the same procedure as the previous experiments.

5.3.3. Experiment evaluation

The evaluation of the previously described experiments is based on a frame by frame evaluation. Since a feature vector is extracted from each frame, the identification system considers each frame as a testing utterance and compares it with all the speaker models to determine the closest speaker match.

The identified speaker at the frame level is compared to the actual speaker of the test utterance and the number of frames that have been correctly identified are recorded to determine the identification rate. See **equation (5.3).**

$$\text{Identification rate (\%)} = \frac{\text{number of correctely identified frames}}{\text{total number of frames}} \times 100 \qquad (5.3)$$

This procedure is repeated for all the set of speakers and their average is tabulated in the experiment results.

- **Evaluation Algorithm:**

The main goal of speaker identification algorithm is to calculate the likelihood of a given test utterance with respect to each speaker model and the one that registered the maximum likelihood is identified as the actual speaker.

Given a test utterance, a sequence of T feature vectors is extracted:

$$X = \{\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots \dots, \vec{x}_T\}$$

Choosing a group S of 10 speakers 10 variables need to be declared and initialized to count the number of times a speaker has been identified :

$$n_1 = n_2 = \dots = n_{10} = 0$$

Step 1: The likelihood of a feature vector is computed with respect to each speaker's model as follows:

$$l_1 = \log P(\vec{x}_1 | \lambda_1)$$

$$l_2 = \log P(\vec{x}_1 | \lambda_2)$$

. . . .

. . . .

. . . .

$$l_{10} = \log P(\vec{x}_1 | \lambda_8)$$

Step 2 : Compare the 10 likelihoods, the maximum one will determine the identity of the speaker:

l= max ($l_1, l_2, \ldots\ldots, l_{10}$)

Step3:

If l=l_1 , speaker 1 has been identified so n_1=n_1+1

If l =l_2 , speaker 2 has been identified and n_2=n_2+1

The same incrementation method is applied for all the 10 speakers

Step 4:

Repeat the previous steps for all the T feature vectors

Step5:

For all the test utterance the identification rate (%) is computed for each speaker

$idr_1 = \frac{n1}{M} \times 100\%$

$idr_2 = \frac{n2}{M} \times 100\%$

. . . . .

. . . . .

$$idr_{10}=\frac{n10}{M}\times 100\%$$

5.3.4. Results and discussion

This section describes the three conducted experiments

5.3.4.1. Experiment 1:

Purpose: Demonstrate the effect of varying the Gaussian model order and the dimensionality of feature vectors on the speaker identification performance.

The results of the first experiment are shown in Table1 such that the measurement are evaluated on a raw speech signal that didn't submit any preprocessing. Random speaker set were selected from TIMIT.

Model order	Identification rate (%)	
	13 MFCC	**39** MFCC
16	57.83	61.95
20	57.48	59.54
24	56.85	57.64

Table 1 Results of the first speaker identification experiment

Different observations are drawn from the tabulated results in **Table1.**

The first one is the effect of varying the number of mixture components has on the identification rate such that a 16 order model seem to give the best results compared to the 20th and 24th order, However the difference is 0.35% and 0.98% respectively which is not really significant .

The second observation concerns the dimensionality variation of the feature vectors which registered an increase up to 4.12% in the identification rate, for higher dimension features.

5.3.4.2. Experiment2:

Purpose: This experiment intends to reveal the effect of increasing the duration of the training data by optimizing the chosen set of speakers available on TIMIT, taking the ones who recorded the longest utterances.

Model order	Identification rate (%)	
	13 MFCC	**39** MFCC
16	60.74	63.42
20	59.86	62.10
24	59.70	60.57

Table 2 Results of the second speaker identification experiment

Table 2 confirms the observations of the previous experiment, such that the 16 component GMM gives the best identification rate and 39 MFCC outperforms the 13MFCC for all the model orders that have been selected .A comparison of experiment 1 and experiment 2 results will be found in **Table 5**.

The best result obtained from the second experiment which is the 16 GMM that uses 39MFCC features, is developed in a confusion table, Table3, with the objective to reveal the detailed computations that led to the final results of the identification experiments.

Hypothesized Speaker Index	Actual Speaker index									
	1	2	3	4	5	6	7	8	9	10
	Identification rate (%)									
1	**70.80**	8.63	3.40	1.60	2.10	3.36	1.27	2.89	1.45	0.78
2	9.13	**68.14**	8.51	3.20	1.45	2.04	2.08	2.98	3.23	1.34
3	5.14	6.86	**71.06**	1.39	1.58	2.16	1.36	1.80	0.22	1.57
4	1.35	1.86	2.78	**52.24**	1.58	3.96	8.60	6.11	5.91	1.57
5	3.38	1.76	1.70	4.06	**74.90**	8.15%	4.34%	5.33	10.03	5.03
6	3.72	1.76	2.24	8.76	4.73	**59.35**	4.25	4.78	5.80	15.88
7	0.78	0.49	3.41	6.94	3.15	5.76	**61.99**	6.97	8.69	3.69
8	4.28	9.22	4.12	11.00	3.15	7.55	10.41	**63.12**	6.24	5.37
9	0.45	0.69	1.97	7.59	4.21	5.04	4.89	4.62	**54.40**	6.60
10	0.68	0.59	0.81	3.21	3.15	2.64	0.81	1.41	4.01	**58.17**

Table 3 Confusion table of 16 order GMM and 39 MFCC coefficients

This table explains the procedure of speaker identification task. For example by selecting a test utterance that belongs to the first speaker (actual speaker) it shows that the system identifies speaker1 (hypothesized speaker) with a percentage of 70.80% whereas the remaining 9 speakers are identified with small percentages such that the highest one recorded is 9.13% which identifies speaker2.

This is an important observation, because it shows that if the measurements were performed on the whole test sentences instead of a frame by frame evaluation, the speaker identification system will take in consideration the highest percentage obtained across the ten speakers and presume that it's the actual speaker . The diagonal values of table 3 record the highest percentages, meaning that if the testing was evaluated along the whole test sentence, the identification rate will be of 100%.

5.3.4.3. Experiment 3:

Purpose: The objective of this experiment is to evaluate the performance of the speaker identification system by taking into consideration only the voiced part of the speech signal, removing silence and the unvoiced segments from both training and testing data. An investigation about decreasing the number of GMM components is also

conducted since it has been observed from the previous experiment that the lowest model order gave the best results.

The results of the experiment are tabulated in **Table 4** such that the measurements on (12, 16, 20) GMM components are performed.

Model order	Identification rate (%)	
	13 MFCC	**39** MFCC
12	72.58	74.09
16	73.01	78.21
20	70.47	67.77

Table 4 Results of the third speaker identification experiment

The first remark of this table is that the 16 order GMM with 39 features still outperforms the other model orders even for the 12^{th} components GMM that has been introduced in this experiment, which confirms that the best identification rate is achieved for a 16 order GMM.

The second observation is that increasing the dimension of MFCC didn't have the same effect on the 20^{th} component this time, since it decreased the performance of the system from 70 to 67% which is not the case for the 16 and 12^{th} model orders which keeps their good performance for higher feature dimensions.

The final remark suggest that lower model orders seem to give better results than the higher ones such that 12 components is preferred to the 20 components GMM.

5.3.4.4. Comparison and interpretation of the results:

Model order	Experiment	Training duration	Speech signal	Identification rate (%)	
				13 MFCC	**39** MFCC
12	3	30sec	Segmented	72.58	74.09
16	1	20 sec	Non segmented	57.83	61.95
	2	30 sec	Non segmented	60.74	63.42
	3	30sec	Segmented	73.01	78.21
20	1	20 sec	Non segmented	57.48	59.54
	2	30 sec	Non segmented	59.86	62.10
	3	30 sec	Segmented	70.47	67.77
24	1	20 sec	Non segmented	56.85	57.64
	2	30 sec	Non segmented	59.70	60.57

Table 5 Comparison of the identification experiments results

This table compares the experimental results obtained in all the speaker identification experiments and summarize all the important information that need to be retained.

The optimal speaker identification system is obtained using 16 components GMM with 13 MFCC dimensions for a training and testing data that have been subjected to preprocessing or a segmentation that kept only the voiced segments of the speech.

The results suggest clearly that preprocessing the data before the feature extraction phase in training and testing, increases significantly the identification rate especially for the 16th component GMM that registered and increase of almost 15%.

Another important observation is the impact of MFCC dimension on the results. In exception to the 20 component GMM, all the results indicate an improvement in the identification rate when using higher dimension (39) MFCCs, however the highest increase recorded is 5.2% for the 16 order GMM, which is not significantly high.

A comparison between the 1st and 2nd experiment shows with no surprise that increasing the training duration plays an important role in improving the identification

rate of the system .The results indicate that increasing the training duration by 10 seconds improved the performance of the system by an average of 3%.

Finally the last observation that can be drawn from the table is that lower order GMMs tend to give better identification rates, such that 12^{th} order GMM outperforms the 20^{th} order which by its turn performs better than the 24 components GMM.

For the verification experiments the segmented and non-segmented data with 39 MFCC will be used and compared.

5.4. Speaker verification:

In this part of this project, the speaker GMMs that have been evaluated in the speaker identification experiments will be used for a text-independent speaker verification task.

Two main experiments will be performed and compared to see the effect of speech segmentation on the verification task.

The next section will give a brief introduction on the speaker verification task, and then the experiments description and evaluation will be covered.

5.4.1. Speaker verification overview:

Speaker verification task consist mainly of a system that takes a binary decision, whether an input utterance belong to the claimed speaker or not.

So for a given input utterance and a claimed identity, the choice becomes:

H_0: if X is from the claimed speaker
H_1:if X is not from the claimed speaker

The general procedure in any speaker verification task is to apply a likelihood-ratio test to an input utterance and determine if the claimed speaker is accepted or rejected.

The likelihood-ratio test is computed with respect to two GMM models. The first model is the model of the claimed speakerλ_c, whereas the second one models a set of possible imposter (non-claimant) speakers and it is known as the background model$\lambda_{\bar{c}}$.

The likelihood-ratio is:

$$\frac{\Pr(\text{X is from the claimed speaker})}{\Pr(\text{X is not from the claimed speaker})} = \frac{\Pr(\lambda_c|X)}{\Pr(\lambda_{\bar{c}}|X)} \quad (5.4)$$

By applying the Bayes' rule and discarding the constant prior probabilities for claimant and impostor speakers, the likelihood ratio in the log domain becomes:

$$\Lambda(X) = \log P(X|\lambda_c) - \log P(X|\lambda_{\bar{c}}) \quad (5.5)$$

such that : $P(X|\lambda_c)$: The likelihood that X belongs to the claimed speaker

$P(X|\lambda_{\bar{c}})$: The likelihood that X does not belong to the claimed speaker

The likelihood ratio Λ is compared with a threshold value θ and the decision becomes:

$$\text{If } \begin{cases} \Lambda(X) \geq \theta & \text{Accept the claimant speaker} \\ \Lambda(X) < \theta & \text{Reject the calimant speaker} \end{cases} \quad (5.6)$$

The decision threshold θ selection is very important, because a high threshold will increase the percentage of false acceptance (FA) error whereas a small threshold imply a higher percentage of false rejection (FR) error, so an adjustment between the trade-off that exist between FA and FR need to be found by optimizing θ.

The terms of the likelihood ratio are computed as follows:

$$\log P(X|\lambda_c) = \frac{1}{T}\sum_{t=1}^{T} \log p(x|\lambda_c) \quad (5.7)$$

The $\frac{1}{T}$ factor is used to normalize the likelihood for the utterance duration.

The likelihood of imposter speakers is formed by using a set of B background-speaker models, $\{\lambda_1, \lambda_2, \ldots, \lambda_B\}$ and $P(X|\lambda_{\bar{c}})$ is the joint probability density that the utterance [5] comes from a background speaker if we assume equally likely speakers :

$$\log P(X|\lambda_{\bar{c}}) = \log \{\frac{1}{B}\sum_{b=1}^{B} p(X|\lambda_b)\} \quad (5.8)$$

Such that $p(X|\lambda_b)$ is computed as in Equation (5.7) by omitting the factor $\frac{1}{T}$

From this overview it is concluded that any speaker verification system face the problem of background speakers selection and threshold selection, such that the

background speakers need to represent the characteristics of all the possible imposters that can be presented to the system.

5.4.2. Speaker verification algorithm:

For this project the speaker verification task is applied on the same set of speakers that as Experiment 2 and 3 of speaker identification, such that the speaker that recorded the highest identification rate is taken as the claimed speaker, whereas the others are selected as background speakers.

A likelihood estimation with respect to the claimant speaker model and background-speakers model is performed on each frame of the input utterance.

Given a sequence of T feature vectors for the claimant test utterance: X= $\{\vec{x}_1, \vec{x}_2, \ldots \vec{x}_T\}$

Step 1:

Compute the likelihood of a feature vector with respect to the claimant and imposter models as follows

l_1= log P ($\vec{x}_1|\lambda_C$)

l_2=log $\{\frac{1}{9}\sum_{b=1}^{9} P(\vec{x}_1|\lambda_b)\}$

Step2:

Compute the likelihood ratio: $\Lambda(X) = l_1 - l_2$ and select a threshold value θ

Step 3:

Compare the likelihood ratio and the threshold value:

$$\text{If } \begin{cases} \Lambda(X) \geq \theta & n1 = n1 + 1 \\ \Lambda(X) < \theta & n2 = n2 + 1 \end{cases}$$

Step 4:

Repeat the procedure for all the feature vectors

Step 5:

Compute FA and FR percentage error:

- If the test utterance belongs to the claimant speaker then: $FR=\frac{n2}{T} \times 100$
- To compute FA, a test utterance from each of the background speakers is taken and for each one a false acceptance percentage is computed as $FA_b = \frac{n1_b}{T} \times 100$

  the total false acceptance is computed as their average : $FA = \frac{1}{9}\sum_{b=1}^{9} FA_b$

Step 6:

Repeat the procedure for different values of θ until ($\frac{FA+FR}{2}$) is minimum

5.4.3. Experiments description and evaluation

First experiment:

The optimized dataset of the speaker identification experiment will be used for the verification task, such that a test utterance of the first speaker will be used as the claimed speaker, whereas the 9 other speakers will be considered as background speakers.

The experiment will be conducted on raw speech signals without any preprocessing and an optimized threshold value will be estimated to minimize the false rejection (FA) of the claimed speaker and the false acceptance of an imposter speaker (FR)

Second experiment

This experiment will follow the same procedure as the first one except that the speech signal will be segmented and only the voiced parts will be considered.

5.4.4. Results and discussion

5.4.4.1. Experiment 1

The results of the first experiments are tabulated in Table 6 where multiple threshold values were tested to obtain the optimum one and the speech signal is non- segmented

θ	FR (%)	FA (%)	$(\frac{FA+FR}{2})$ (%)
0.1	25.03	4.77	14.90
0.5	25.48	4.26	14.87
1	26.83	3.73	15.28
1.5	27.85	3.15	15.50
2	29.09	2.69	15.89

Table 6 First speaker verification experiment

As expected it is observed in **Table 6** that FR and FA are inversely proportional such that when the threshold increases FR increases and FA decreases.

Another important observation is that$(\frac{FA+FR}{2})$ is minimum for θ=0.5, and this value is surrounded by a smaller and higher threshold that registered a higher average error. This suggest that the optimal θ can be found in this range: $0.1 < \theta_{optim} < 1$

Another range of threshold is selected in Table7 to see if a better threshold can be found.

θ	FR (%)	FA (%)	$(\frac{FA+FR}{2})$ (%)
0.2	25.14	4.64	14.89
0.3	25.25	4.52	14.89
0.4	25.48	4.37	14.93
0.6	25.70	4.21	14.96
0.7	26.27	4.13	15.20
0.8	26.38	4.01	15.20
0.9	26.49	3.86	15.18

Table 7 Threshold selection for the first speaker verification experiment

The smallest average error that is measured in Table7 is 14.89% for a two different threshold values of 0.2 and 0.3 ,However the obtained result in Table 6 shows a slightly smaller error percentage of 14.87 for θ= 0.5. So even if we took a smaller range to determine more accurately the optimum threshold, it is confirmed that the first result of Table 6 assures a minimum average error.

5.4.4.2. Experiment 2

In this part, the data-set of speakers is unchanged and the same procedure as the previous experiment is followed, however the speech signal has been segmented before the feature extraction and only the voiced parts are kept. The results are shown in Table 8

θ	**FR (%)**	**FA (%)**	**$(\frac{FA+FR}{2})$ (%)**
0.1	6.86	2.67	4.77
0.5	7.07	2.49	4.78
1	7.9	2.27	5.09
1.5	8.73	2.05	5.38
2	9.56	1.83	5.70

Table 8 Second speaker verification experiment

Taking the same threshold values as in the first experiment, the smallest average error shown in Table 8, is recorded for the first value of θ = 0.1 meaning that another range should be taken into consideration and smaller values should be investigated.

Table 9, aims to find an optimum threshold value:

θ	FR (%)	FA (%)	$(\frac{FA+FR}{2})$ (%)
0.2	6.86	2.59	4.73
0.3	7.07	2.56	4.82
0.4	7.07	2.52	4.80

Table 9 Threshold selection for the second speaker verification experiment

This table suggest that the optimum threshold value for the segmented speech signal is θ=0.2 with a minim error of 4.73%

5.4.4.3. Comparison and interpretation:

θ	Speech signal	FR (%)	FA (%)	$(\frac{FA+FR}{2})$ (%)
0.1	Non-segmented	25.03	4.77	14.90
	Segmented	6.86	2.67	4.77
0.2	Non -segmented	25.14	4.64	14.89
	Segmented	6.86	2.59	**4.73**
0.3	Non-segmented	25.25	4.52	14.89
	Segmented	7.07	2.56	4.82
0.4	Non -segmented	25.48	4.37	14.93
	Segmented	7.07	2.52	4.80
0.5	Non -segmented	25.48	4.26	**14.87**
	Segmented	7.07	2.49	4.78
1	Non -segmented	26.83	3.73	15.28
	Segmented	7.9	2.27	5.09
1.5	Non -segmented	27.85	3.15	15.50
	Segmented	8.73	2.05	5.38
2	Non -segmented	29.09	2.69	15.89
	Segmented	9.56	1.83	5.70

Table 10 Results comparison of speaker verification experiments

Table 10 summarize the obtained results in speaker verification experiments. From the 1st experiment, the optimum threshold θ=0.5 had recorded a minimum error of 14.87%, whereas the 2nd experiment recorded a minimum error of 4.73% for a smaller optimum threshold of θ=0.2.

Another important observation that is seen in Table 10 is that the main improvement concerns the minimization of the false rejection error that has been reduced by almost 20% when the speech signal was segmented.

These results demonstrate clearly that preprocessing the speech signal improve significantly the performance of the speaker verification system such that it minimized the false acceptance (FA) and false rejection (FR) error average by 10% .

It is important to notice that the speaker verification results were based on a frame by frame basis and since the minimum average error is 4.73%, this means that the claimed speaker has been accepted with a percentage of 95% and if the system takes the whole testing utterance sentence it will apply a majority rule over all the number of frames and the verification will be of 100%.

Chapter 6

Conclusion

6.1. Conclusion

This project presented a text-independent speaker recognition system that uses GMM to model a set of speakers from the TIMIT database. The speaker identification and speaker verification experiments had the objective to show the important parameters that can improve the system's recognition performance.
For that purpose different components GMMs have been modeled and two distinct feature's dimension have been used. The results suggest that the optimum GMM order is 16 and 39 MFCC improve slightly the performance of the identification task compared to the 13MFCC coefficients.
The other important observation that is drawn is the importance of preprocessing (segmenting) the speech signal before the feature extraction phase, such that by using a simple segmentation method that is based on short-time energy estimation and the zero crossing rate , the identification rate of the system increased by 15%.

It is also important to notice that TIMIT database provide a small duration utterances for each speaker and all the obtained results were based on a training that uses a maximum of 30seconds speech data, which is a quite small duration for a speaker recognition system. Nevertheless satisfactory results have been obtained from the multiple experiments.

6.2. Suggestion for further research:

To improve the performance of the developed speaker recognition system, the most obvious work that need to be done is to to invest more on the preprocessing phase of the speech signal, such that using powerful preprocessing method will contribute to significant increase of the identification rate .

To get better results we can also try to combine different types of features that can capture different characteristics of a speaker's voice, leading to more optimized speaker models

Finally we can think of using different speaker based database that have longer durations of speaker recordings like VoxCeleb [54] and YOHO [55, 5] database, specially designed for speaker recognition tasks.

References

[1] G. A. v. Graevenitz, «About Speaker Recognition Technology,» Bonn-Germany.

[2] V. Voicevault, «Voicevault-a brief history of voice biometrics,» 02 september 2015. [En ligne]. Available: https://voicevault.com/a-brief-history-of-voice-biometrics/. [Accès le 23 06 2019].

[3] L. L. T.F. Zheng, «Robustness-Related Issues in Speaker Recognition,» *SpringerBriefs in Signal Processing,* 2017.

[4] Y. D. Jin.Minho, «Speaker Verification and Identification,» korea.

[5] D. A.Reynolds, «Automatic Speaker Recognition using Gaussian Mixture Speaker Models,» *The Lincoln Laboratory Journal ,* vol. 8, n° %12, pp. 173-191, 1995.

[6] S. Ozaydin, «Design of a text independent speaker recognition system,» chez *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ankara, 2017.

[7] «Wikipedia-Speaker recognition,» [En ligne]. Available: https://en.wikipedia.org/wiki/Speaker_recognition. [Accès le 23 06 2019].

[8] S. Furui, «50 Years of Progress in Speech and Speaker,» *ECTI TRANSACTIONS ON COMPUTER AND INFORMATION TECHNOLOGY,* vol. 1, n° %12, pp. 64-74, November 2005.

[9] S. Pruzansky, «Pattern-matching procedure for automatic talker recognition,» *J.A.S.A,* vol. 35, pp. 354-35, 1963.

[10] G. R. Doddington, «A method of speaker verification,» *J.A.S.A,* vol. 49, n° %1139 , 1971.

[11] e. a. W. Endress, «''Voice spectrograms as a function of age'',''Voice Disguise and Voice Imitation'',» *J.A.S.A,* vol. 6, n° %12, pp. 1842-1848, 1971.

[12] S. Furui, «An analysis of long-term variation of feature parameters of speech and its application to talker recognition,» *Electronics and Communications in Japan,* vol. 57, n° %1A, pp. 34-41, 1974.

[13] L. E. Baum et T. Petrie, «Statistical Interference for Probabilistic Functions of Finite State Markov Chains,» *The Annals of Mathematical Statistics,* vol. 37, n° %16, pp. 1554-1563, 1966.

[14] e. a. F. K. Soong, «A vector quantization approach to speaker recognition,» *AT&T Technical Journal,* vol. 66, pp. 14-26, 1987.

[15] D. A. R. a. R. C. Rose, «Robust Text-Independent Speaker Identification,» *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING,* vol. 3, n° %11, pp. 72-83, January 1995.

[16] T. M. a. S. Furui, «Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMMs,» *Proc. ICSLP,,* vol. 2, pp. 157-160, 1992.

[17] R. R. S. Nilu Singh, «Applications of Speaker Recognition,» *Procedia Engineering ,* vol. 38, pp. 3122-3126, 2012.

[18] B. W. G. C. M. PRIYANKA A. ABHANG, Introduction to EEG and Speech based Emotion Recognition, Aurangabad: Elsevier Inc, 2016.

[19] E. Lai, «Speech synthesis,» chez *Practical Digital Signal Processing for Engineers and Technicians*, IDC Technologies, 2003, pp. 88-90.

[20] P. Sjölander, «Research Gate,» 8 August 2012. [En ligne]. Available: https://www.researchgate.net/post/How_would_you_define_Speech_Perception. [Accès le 21 juin 2019].

[21] [En ligne]. Available: https://www.d.umn.edu/~jfitzake/Lectures/DMED/InnerEar/ExtMidEar/EarAnatomy.html.

[22] «ASHA,» [En ligne]. Available: https://www.asha.org/public/hearing/How-We-Hear/. [Accès le 21 06 2019].

[23] «Wikipedia- speech processing,» [En ligne]. Available: https://en.wikipedia.org/wiki/Speech_processing. [Accès le 21 06 2019].

[24] J. JOSEPH P. CAMPBELL, «Speaker Recognition: A Tutorial,» *PROCEEDINGS OF THE IEEE,* vol. 85, n° %19, pp. 1437-1462, 1997.

[25] «nipunbatra-github,» [En ligne]. Available: https://nipunbatra.github.io/blog/2014/dtw.html. [Accès le 22 06 2019].

[26] «Wikipedia-DTW,» [En ligne]. Available: https://en.wikipedia.org/wiki/Dynamic_time_warping. [Accès le 22 06 2019].

[27] G. Z. A. S. A. M. Abdelmajid H. Mansour, «Voice Recognition using Dynamic Time Warping and Mel-Frequency Cepstral Coefficients Algorithms,» *International Journal of Computer Applications,* vol. 116, n° %12, p. 0975 – 8887, April 2015.

[28] S. Dorairaj, «Media-Hidden Markov Models Simplified,» 20 March 2018. [En ligne]. Available: https://medium.com/@postsanjay/hidden-markov-models-simplified-c3f58728caab. [Accès le 22 06 2019].

[29] «Practical Cryptography -Hidden Markov Model (HMM) Tutorial,» [En ligne]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/hidden-markov-model-hmm-tutorial/.

[30] S. M. A. ,. E. H. ,. R. Alaa Ehab Sakran, «A Review: Automatic Speech Segmentation,» *International Journal of Computer Science and Mobile Computing,* vol. 6, n° %14, pp. 308-315, April 2017.

[31] K. S. B. B. B. Bachu R.G, «Separation of Voiced and Unvoiced using Zero crossing rate and Energy of the Speech Signal,» chez *ASEE Regional Conference*, Bridgeport, March 2008.

[32] «Feature (machine learning),» 10 09 2018. [En ligne]. Available: https://en.wikipedia.org/wiki/Feature_(machine_learning).

[33] D. M. D. Mr.Yoghesh Dawande, «ANALYSIS OF DIFFERENT FEATURE EXTRACTION TECHNIQUES FOR SPEAKER RECOGNITION SYSTEM: A REVIE,» *International Journal of Advanced Technology & Engineering Research (IJATER,* vol. 5, n° %11, pp. 5-7, Jan 2015.

[34] N. Dave, «Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition,» *INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY ,* vol. 1, n° %1VI, July 2013.

[35] «Wikipedia,» March 2010. [En ligne]. Available: https://en.wikipedia.org/wiki/Linear_predictive_coding. [Accès le 30 05 2019].

[36] W. K. K.K. Paliwal, «Quantization of LPC Parameters».

[37] H. Hermansky, «Perceptual linear predictive (PLP) analysis of speech,» *J. AcousL Soc. Am,* vol. 87, n° %14, pp. 1738-1752, April 1990.

[38] Davis, S., Mermelstein et P., «Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,» *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 28, n° %14, pp. 357-366, 1980.

[39] S. S. Stevens, J. Volkmann et a. E. B. Newman, «A Scale for the Measurement of the Psychological Magnitude Pitch,» *Journal of the Acoustical Society of America,* vol. 8, n° %13, pp. 185-190, 1936.

[40] «practical cryptography,» 2009. [En ligne]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/.

[41] P. Nair, «Medium,» 24 Jul 2018. [En ligne]. Available: https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd. [Accès le 2019].

[42] A. oppenheim et R.W.Schafer, «From frequency to quefrency: a history of the cepstrum,» *IEEE Signal Processing Magzine ,* pp. 95-106, september 2004.

[43] S. K. K. a. M. Laxminarayana, "CHOICE OF MEL FILTERBANK IN COMPUTING MFCC OF A RESAMPLED SPEECH," Mumbai, MAY 2010.

[44] T. Gisselbrecht, «medium,» 2 may 2018. [En ligne]. Available: https://medium.com/snips-ai/machine-learning-on-voice-a-gentle-introduction-with-snips-personal-wake-word-detector-133bd6fb568e. [Accès le 16 april 2019].

[45] S. M. A. G. Md. Afzal Hossan, «A Novel Approach for MFCC Feature Extraction,» chez *4th International Conference on Signal Processing and Communication systems* , Melbourne , 2010.

[46] A. S. a. S. K. Singla, «State-of-the-art Modeling Techniques in Speaker,» *International Journal of Electronics Engineering ,* vol. 9, n° %12, pp. 186-195, 2017.

[47] G. P. V. T. John McGonagle, «Brilliant,» [En ligne]. Available: https://brilliant.org/wiki/gaussian-mixture-model/. [Accès le 12 april 2019].

[48] A. Aroon et S.B.Dhonde, «Speaker Recognition System using Gaussian Mixture,» *International Journal of Computer Applications,* vol. 130, n° %114, pp. 38-40, November 2015.

[49] J. A. Bilmes, «A Gentle Tutorial of the EM Algorithm And its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,» International Computer Science Institute, Berkeley CA, 94704, 1998.

[50] M. I. Jordan et R. A. Jacobs, «Hierarchical mixtures of experts and the EM algorithm,» chez *International Joint Conference on Neural Networks*, 1993.

[51] A. P. Dempster, N. M. Laird et D. B. Rubin, «Maximum Likelihood from Incomplete Data via the EM Algorithm,» *Journal of the Royal Statistical Society,* vol. 39, n° %11, pp. 1-38, 1977.

[52] Z. Hu, «Initializing the EM Algorithm for Data Clustering and Sub_population Detection,» Ohio State University , 2015.

[53] «Linguistic data consortium,» [En ligne]. Available: https://catalog.ldc.upenn.edu/LDC93S1.

[54] VoxCeleb. [En ligne]. Available: http://www.robots.ox.ac.uk/~vgg/data/voxceleb/.

[55] «Linguistic data consortium,» [En ligne]. Available: https://catalog.ldc.upenn.edu/LDC94S16.

[56] SadaokiFurui, «Speaker Recognition in Smart Environments,» chez *Human-Centric Interfaces for Ambient Intelligence*, Tokyo, Academic Press, 2010, pp. 163-184.