الجامعة البريطانية في دبي

The British University in Dubai

A Study on Speaker Recognition System

دراسة حول نظام التعرف على المتحدث

By

Hazem Wa’il Mohammed Bakkar

2013128078

Dissertation submitted in partial fulfillment of

MSc Informatics (Knowledge and Data Management)

Faculty of Engineering & Information Technology

Dissertation Supervisor

Prof. Khaled Shalaan

May-2015

DISSERTATION RELEASE FORM

Student Name	Student ID	Programme	Date
Hazem Wa'il Mohammad Bakkar	**2013128078**	**Informatics (Data and Knowledge Management)**	**31.May.2015**

Title

A Study on Speaker Recognition System

I warrant that the content of this dissertation is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that one copy of my dissertation will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make that copy available in digital format if appropriate.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my dissertation for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

Signature

Abstract

The huge development in information technology opened the door for finding an increasing number of security gaps in the daily used systems like email accounts. Security systems developers and manufacturers are trying hardly to cope with the increasing security breaching attacks. The need to overcome this challenge forced many researchers and manufacturers to think about adding extra levels of security to protect information and resources; these extra levels of security are mainly involve around using the human biometrics in order to identify the real identity of the user. Speaker recognition methods are considered a leading approach in applying biometric security systems.

In this thesis we aimed to develop a unique speaker recognition system with a user friendly interface. The proposed system was mainly developed using Python (Python.org, 2015). This system was used to implement and study several methods and techniques in speaker recognition domain.

Another main goal for conducting this research is to make a scientific comparison between tools and methods that are related to speaker recognition domain, the following are the techniques that were studied : 1) Energy based tool and Long-Term Spectral Divergence (LTSD) in the preprocessing module of the system, 2) Mel Frequency Cepstral Coefficients (MFCC) and Linear Predictive Cepstral Coefficients (LPCC) in the feature extraction module, and 3) scikit-learn Gaussian Mixture Model (GMM), Universal Background Model (UBM), Continuous Restricted Boltzmann Machine (CRBM) and Joint Factor Analysis (JFA) in the recognition module. Finally, we proposed a new GMM in this thesis which was compared with the famous scikit-learn GMM technique.

All the mentioned tools and methods were tested and experimented in this thesis. Findings of the experiments showed that: 1) LTSD for voice activity detection is faster and more practical than the energy based tools, 2) MFCC is computationally more expensive than LPCC but MFCC is faster and more accurate, also LPCC needs double size utterance to achieve the same accuracy MFCC generates. 3) The new GMM showed that it is five times faster than scikit-learn GMM, also the proposed GMM outperforms all other techniques studied in this thesis. As a result, to build a user-friendly speaker recognition system, it is better to use LSTD for preprocessing, MFCC for feature extraction, and our enhanced GMM for speaker testing and recognition.

خـلاصــة

إن التقدم الكبير في تكنولوجيا المعلومات فتح الباب لظهور أعداد متزايدة من الثغرات الأمنية في أنظمة المعلومات يومية الاستخدام، مثل حسابات البريد الالكتروني. و هذا بدوره جعل مصنعي و مطوري أنظمة الأمن و الحماية يحاولون جاهداً مجاراة الهجمات المتزايدة التي تتعرض لها أنظمة المعلومات من خلال الاختراقات الأمنية. إن الحاجة المتزايدة للتغلب على هذه التحديات دفعت الباحثين و المصنعين للتفكير بتطوير مستويات أمنية إضافية لحماية المعلومات و الموارد، هذه المستويات الأمنية الاضافية تدور بشكل خاص حول تفعيل دور الخصائص الحيوية الانسانية في أنظمة الحماية، حيث تمتاز الخصائص الحيوية بقدرتها على تحديد هوية المستخدم الحقيقية، و تعتبر أنظمة التعرف على المتحدث من خلال صوته من أنجع أساليب الخصائص الحيوية في تطبيقات الأمن و الحماية التي تعتمد على الخصائص الحيوية.

تهدف هذه الأطروحة إلى تطوير و إنشاء نظام فريد للتعرف على المتحدث سهل الاستخدام. إن النظام المقترح تم تطويره بشكل أساسي باستخدام لغة البرمجة Python، و قد تمت الاستفادة من هذا النظام في دراسة و تطبيق العديد من الطرق و التقنيات المستخدمة في مجال تطوير أنظمة التعرف على المتحدث.

يهدف هذا البحث إلى تحقيق هدف أساسي آخر، و هو القيام بعمل مقارنة علمية بين الأدوات و التقنيات الخاصة في مجال تطبيق أنظمة التعرف على المتحدث، و هذه التقنيات هي: 1) الأداة المعتمدة على الطاقة و الاختلاف الطيفي طويل الأمد (LTSD) 2) معاملات ميل التردد السبسترولي MFCC و معاملات التنبؤ الخطي السبسترولي LPCC في وحدة استخلاص الملامح. 3) نموذج مزيج غاوس الخاص بـscikit-learn و نموذج الخلفية العالمي، ماكينة بولتزمان المستمرة المقيدة، و تحليل العامل المشترك في وحدة التعرف على المتحدث. و أخيراً فقد قمنا بطرح نموذج مزيج غاوس جديد في هذه الأطروحة، و قد تم عمل مقارنة بينه و بين نموذج مزيج غاوس الخاص بـscikit-learn.

كل التقنيات و الطرق المذكورة سابقاً تم اختبار آدائها و تجربتها باستخدام النظام المقترح في هذه الأطروحة, و قد كانت النتائج على النحو التالي: 1) الاختلاف الطيفي طويل الأمد أسرع من الأداة المعتمدة على الطاقة و عملي أكثر في الكشف عن النشاط الصوتي في الإشارة.2) معاملات ميل التردد السبسترولي MFCC أكثر تعقيداً حسابياً من معاملات التنبؤ الخطي السبسترولي LPCC، و لكنها بالرغم من ذلك أسرع و أكثر دقة من معاملات التنبؤ الخطي السبسترولي LPCC التي تحتاج إلى دراسة حديث صوتي حجمه ضعف حجم الحديث الصوتي الذي تحتاجه معاملات ميل التردد السبسترولي MFCC للوصول إلى نفس درجة الدقة. 3) نموذج مزيج غاوس الجديد و المقترح في هذه الأطروحة أسرع خمس مرات على الأقل من نموذج مزيج غاوس الخاص بـscikit-learn، أيضا لقد فاق النموذج المقترح باقي التقنيات التي تمت عليها التجارب و

الاختبارات في هذه الأطروحة. و كنتيجة نهائية للأطروحة فإنه لبناء نظام للتعرف على المتحدث سهل الاستخدام و عملي فإنه من الافضل أستخدام الاختلاف الطيفي طويل الأمد في وحدة التجهيز، و معاملات ميل التردد السبسترولي MFCC في اسخلاص الملامح الصوتية ، و نموذج مزيج غاوس المقترح و المحسن في وحدة التعرف على المتحدث.

Acknowledgements

First of all, I am so thankful and grateful to Allah for everything in my whole life, I am happy that Allah gave me the chance to pursue my education, which was always a superior goal in my life,

I would like to thank my supervisor Prof. Khaled Shaalan for his support and guidance in my dissertation, his enthusiasm leaded me in all my work in this dissertation.

No one can deny the support and efforts Prof.. Khaled and Dr. Sherief Abdalla put for me, both of them facilitated me in my tough experience in travelling between two countries to attend the lectures. Without their support I will not reach this point in my study.

Finally, the great love and patience from my wife, the courage I droned from my parents, and the support from my parents in law and all my family in Dubai, make me strong enough to face all the challenges in my Master study journey.

I dedicate this work to my beloved sweet daughters *Fai* and *Sama*.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Hazem Wa'il Mohammed Bakkar)

Table of Contents

Table of Figures

Chapter 1

Introduction

This Chapter gives an overview about the importance of Speaker recognition systems and the main reasons behind the urgent need of such systems. Additionally, this chapter defines the problem under study and the main goals of conducting this research. The questions that this thesis will answer are also mentioned. Finally, this chapter details the structure of this thesis.

1.1 Overview about Speaker Recognition Systems

In our life, there are many ways humans can communicate with each other, for example, textual language, body language and speech. Speech is considered the most effective and practical way to communicate; because speech contains many types of information within it, such as gender, emotional state, and attitude in addition to the speaker identity. These information are very important and basic for conducting effective communication channels between individuals.

Speech can be represented by a signal carrying information, and this signal can be represented in a wave form that has unique characters. By extracting these characters and analyzing them, we can discover valuable information such as the real identity of the speaker, which is the main subject of this thesis.

Speech signal is rich of information; it mainly reveals three types of vital information: Language identity, speech text and speaker identity as shown in figure (1).

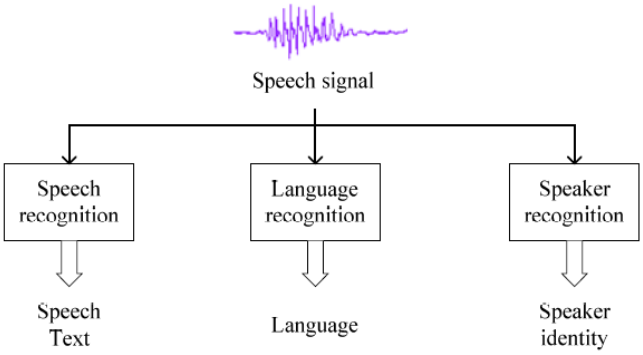


Figure 1: Types of information inside the speech signal (Chang,2015)

Recently, most researches that involved about identifying the speaker identity trends to use automatic speaker recognition systems, which aim to specify the speaker identity using the information lies within the voice signal. These information are extracted then analyzed, the results of this process will specify the speaker identity (Chang, 2015).

Automatic recognition system performs two important tasks; speaker verification and speaker identification. The speaker identification is to specify the identity of a speaker within a known group of speakers. This is done by comparing the voice signal of the speaker with stored voice models in the system; the closest matching model will determine the speaker identity (Chang, 2015).

The process of deciding whether the speaker is the person that he/she is claiming to be, using his/her voice signal is called speaker verification. In

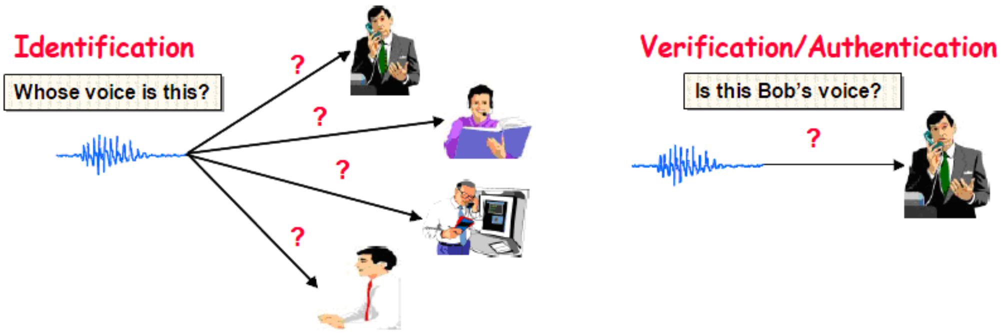


Figure 2: Speaker Identification and Verification Adapted from (Reylonds and Douglas, 2002)

this task the voice signal of the tested speaker is compared to the target voice signal stored in the system, if the likelihood between the two signals exceeds a specified threshold then the tested speaker is accepted, otherwise he/she will be refused. Figure (2) shows these two tasks (Chang, 2015).

1.2 Problem definition

It is common these days to use a combination of a valid username and password to be authenticated using a security system. Using this combination of credentials does not identify the actual user who owns them, anyone who knows the username and password can use them to get authenticated and get access to use the system resources.

To solve this problem, it is necessary to use information that is unique to each user, and cannot be used by other users. Human biometrics- such as the voice print- are unique and cannot be used by other individuals. In this context, many researchers with the help of manufacturers tried to develop speaker recognition systems that identify the user by analyzing his/her unique voice characteristics. The accuracy of these systems depends on the collected voice samples that will be used to train the speaker recognition system. Voice samples usually need to be large, and they are collected by requesting the user to read relatively long text loudly while recording his/her voice. This primary practice gives the user undesirable experience and it does not tempt the users to go through this practice, specially the users who are not patience.

1.3 Goals

The main goal of this research is to develop a speaker recognition application; which will be able to identify the tested speaker achieving the highest possible accuracy percentage and the least user interaction. The system is going to be trained using recorded voice samples of several speakers; the training phase objective is to allow the system to achieve the most accurate results in identifying the tested speaker. The final expected results of the proposed application are: 1) to accurately identify a user after speaking shortly, and 2) to ignore an imposter who is faking his voice sample.

1.4 Questions and ideas that this thesis should address

- Which method is performing better in signal feature extraction MFCC or LPCC?
- Conducting a performance comparison between several speakers modelling methods and finding out which one is the best to implement.
- How can we enhance the performance of the chosen modelling method to boost its output?
- Can Python be used to achieve the main goal of this thesis which is to develop an effective real world speaker recognition system?

1.5 Thesis structure

The rest of this thesis is organized as follows, chapter (2) introduces a deep background about speaker recognition systems, it describes the tools and methods used in the proposed application, also it presents a brief about the previous researches that were conducted on the speaker recognition systems.

In chapter (3) we explain the methodology that was used to design and develop the proposed speaker recognition system, detailing all the methods and tools that were developed in the proposed system.

Chapter (4) contains the implementation of the proposed system depending on the methodology introduced in Chapter (3). Chapter (5) main subject is explaining all the tests and experiments conducted on the proposed system using different methods related to speaker recognition domain. This chapter also details the results of these experiments and summarizes the performance level achieved in each experiment and test. Finally, Chapter (6) presents the conclusions extracted from this thesis, also it suggests ideas that can be pursued in the future as an extension of this thesis scope and problem.

Chapter 2

Background

This chapter summarizes the main concepts that are necessary for the reader to understand this thesis, also it describes the main tools and methods that are used in speaker recognition systems.

2.1 Overview about Speaker Recognition System

Speaker recognition system is any system that tries to identify a user by studying his voice biometric features. It tries to identify a speaker by building a model that represents the vocal characteristics of a person, this model can be one of two types of models: a statistical model that produces a similar output as the human voice tract, or it could be a mathematical model that produces a human speech (Miller, 1959) (Flanagan, 1972). After creating a model and linking this model to a person, speaker recognition system compares the generated model with a new observed model, and depending on the degree of likelihood between these models the system recognizes a stored voice (Beigi, et al,1998), this is the main methodology that is used in most of the voice recognition systems.

2.2 Why human voice is unique?

The voice generated by the larynx is unique and cannot be the same voice that is generated by another human; this is because every human being has his own unique Anatomical and behavioral vocal features and characteristics. The anatomical characteristics like the size of the larynx which in males is bigger than females causes a difference between males and females voices in term of tone and frequency (Campbell, 1997). On the other hand, the behavioral characteristics which are called also behaviometrics (CHATZARAS & SAVVIDIS, 2015) shape the uniqueness of the human voice; these characteristics include the emotion state of the human, the way the mouth is moving when the voice is generated and the pronunciation under different emotional states (CHATZARAS & SAVVIDIS, 2015).

2.3 Speaker recognition and speech recognition

Usually people confuse between the two terms: speech recognition and speaker recognition, each one of them uses a totally different technology. Speech recognition is identifying what is said by a person while speaker recognition is identifying who is this person. A well-known example of speech recognitionapplication is the service Siri which is used in iPhones. On the other hand, Recgnito is considered a well-known example of a speaker recognition system (Anon, 2015).

2.4 Speaker recognition types:

There are two types of speaker recognition systems in regard to the content that these systems can process (CHATZARAS & SAVVIDIS, 2015) (Sánchez, 2010):

1- Text dependent speaker recognition.
   In this type, the system is trained using a sequence of words or sometimes only a one word, and in the recognition phase the user should say and repeat the same words that were used in training the system in order to recognize the user.
2- Text independent speaker recognition.
   In this type of systems there is no need for saying or speaking the same words used in the training phase, this type of systems usually need longer time of utterance in the training phase, and also longer utterance in the recognition phase compared to the text-dependent systems.

2.5 Overview about Speaker Recognition main processes

Speaker recognition system consists mainly of the following phases (CHATZARAS & SAVVIDIS, 2015) (Sánchez, 2010):

2.5.1 Enrollment phase (Training Phase)

In this phase the system creates a model that describes the user's voice. Usually the system will be trained to identify the user voice, the enrollment phase contains a voice recording process or uploading pre-recorded voices that can be a few seconds long or in some cases hours of speech. The longer the recordings durations, the more information is collected about the user's voice, and this will eventually increase the system accuracy in recognizing the speaker's identity.

2.5.2 Recognition phase (Testing Phase)

The system in this phase captures an unknown voice recording and compares it with the stored models from the previous phase in order to recognize the speaker of this recording. In this comparison the system will calculate the likelihood percentage between the captured voice and the stored models, and depending on the result it will identify whether the unknown voice belongs to one of the registered users or not. The likelihood percentage is determined in the speaker recognition system and it varies from an implementation to another. In our system, we conducted several experiments to choose the most suitable threshold that achieved the best recognition results.

In our system the enrollment phase is completed before starting the recognition phase, but this is not necessary in some systems, recognition phase cannot started before the enrollment phase but it can be performed simultaneously with the enrollment phase, for example there is a well-known voice recognition system called recognito, recognito performs the recognition process simultaneously with the enrollment process (CHATZARAS & SAVVIDIS, 2015) (Sánchez, 2010).

In order to measure the performance of a speaker recognition system, some elements should be considered and processed before measuring the quality of the system's performance; the following summarizes the types of these elements (Alam et al., 2013):

Technical elements: these elements are the ones that determine the technical aspects of the input or output, for example the wave rate of the sampled voice.

Voice capturing elements: these elements describe the effect of the surrounding environment that accompanied the recording session, for example the distance between the speaker and the microphone.

Psychological state of the speaker: this element considers the psychological state of the speaker, when the speaker is under stress he will record a different voice sample than the one when he is extremely happy. This difference in the speaker's state will result a difference in the recorded features for the user's voice.

2.6 Voice Features and characteristics:

In the enrollment phase, the user registers himself in the speaker recognition system, every sample captured from the user's voice is divided into several short frames that range in length between 20-30 milliseconds. The frames then are put under preprocessing phase, any frame that contains weak voice signal will be deleted (Campbell, 1997). After this the unique human voice features and characteristics are extracted from these frames, composing the core of the speaker recognition system which is the unique voiceprint of a specific user. The main extracted voice features are categorized into the following groups (Kinnunen and Li, 2010):

1- Short term spectral features

The short-term spectral features contain information about the user's vocal tract specifications, for example the location of formants in the spectrum, and many other specifications. This information is kept in a so call spectral envelope. Figure (3) shows two different spectral envelopes, one for male and the other for a female.

Short-term spectral analysis is mainly used to build a spectrogram that represents the vocal signal; this analysis is performed by splitting the captured voice signal into frames, each frame duration varies between 20-

30 milliseconds, with a shifting margin that does not exceed 10 milliseconds. The main reason behind splitting the voice signal into frames is that the voice signal is non-stationary, but when it is divided into frames

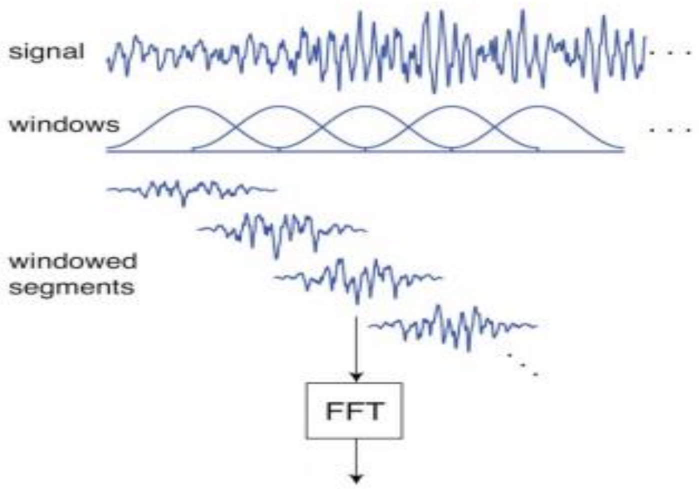


Figure 3: Short-term spectral analysis (Reylonds and Douglas, 2002)

it becomes stationary for a short time. This will give the voice signal the ability to be analyzed. Figure (4) shows the short-term spectral analysis.

2- Spectro- temporal features

Spectro – temporal features are the features extracted from the frequency content of the voice signal spectrogram as shown in figure (5). These features contain information that are essential for recognizing the speaker. Formant transitions and energy modulations are some of the most important features extracted from Spectro-temporal features.

In the last few years, modulation features were proposed and

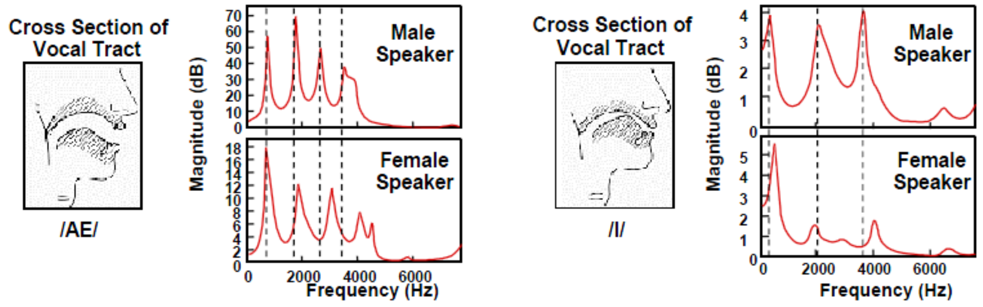


Figure 4: Spectral envelopes one for male and female Adapted from (Chang, 2015)

studied, these features are extracted by representing the non-stationary voice signals as two groups, the first group is Amplitude Modulation (AM) and the second group is the Frequency Modulation (FM).

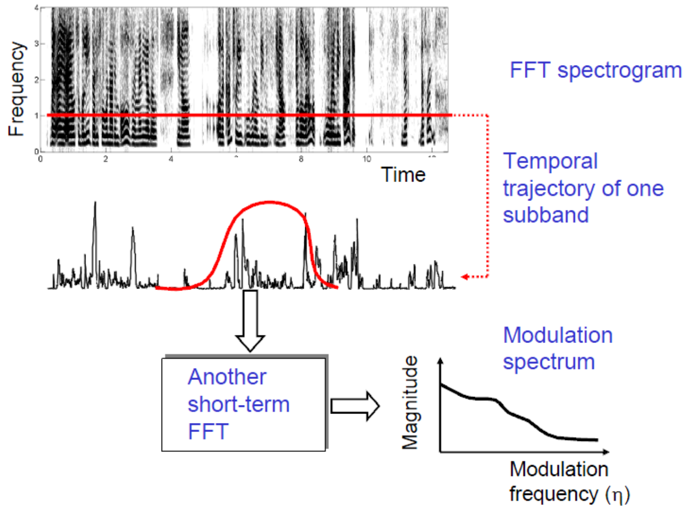


Figure 5: Voice signal spectrogram adapted from (Chang, 2015)

In order to get the most of the temporal information and short-term spectral features, a merge of some of the temporal information with short-term spectral features can be performed by using the first two orders in differences of the feature vectors.

Delta and Delta-Delta Cepstral Coefficients are the names of the first two orders in differences in MFCC, these coefficients are usually added to the original MFCC coefficients at the frame level, for example if we have 12 MFCCs the result of this addition will be extracting about 36 features per

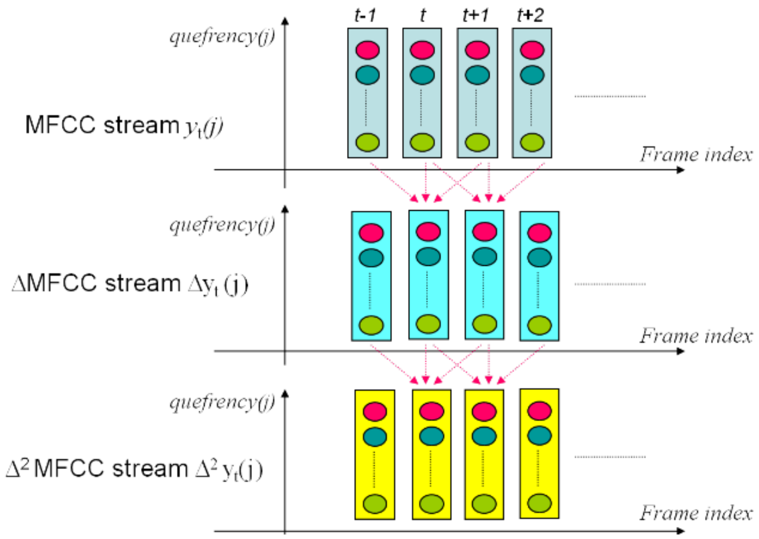


Figure 6: MFCC and the first two order differences adapted from (Chang, 215)

frame. Figure (6) shows MFCC and the first two order differences.

3- Prosodic features

Prosody describes the speaking style, rhythm of speech, and stresses of the words syllable. The features which can be extracted from Prosody contain speaker specific information such as speaking rate and style, language background and emotions that lies inside the speaking style. This type of features usually needs long voice signal segments in order to be extracted, this is because the information reserved in these features are spanned over long segments of voice signals, un-like short-term spectral features which can be extracted from relatively shorter voice signals.

4- Voice source feature

Voice source features contain the information that describes the source in which the voice signal is produced from. Glottal pulse shape and fundamental frequency are considered the most important features of the voice source features. In order to extract this type of features, the researcher should assume that the vocal tract and voice source are independent; because when the voice signal is produced from its source, it should pass the vocal tract; this will cause changes in many of its information, which will eventually lead to major changes on the voice source features.

In order to extract the voice source feature we first estimate the vocal tract filter using linear prediction model, the remain of this process will be the voice source signal which is estimated by reversing the filtering of the voice signal, figure (7) shows an example about the extracting process of the voice source features from a voice signal. The voice source signal is calculated using the following equation (Chang, 2015):

$$E(z) = S(z) \cdot \frac{1}{H(z)} \qquad \textit{Equation (1)}$$

Where $S(z)$ is the voice signal, $H(z)$ is the result of the vocal tract filter and $E(z)$ is the voice source signal.

The voice source features are less sensitive to the speech content than short-term spectral features like MFCC features; because the voice source features are extracted from the voice source signal which depends on the pitch that is produced using the vocal folds. The voice source features are known as not specific or unique like vocal tract features. However, if combined together, short-term spectral features with voice source features, the accuracy of the speaker recognition system will be highly improved and enhanced.

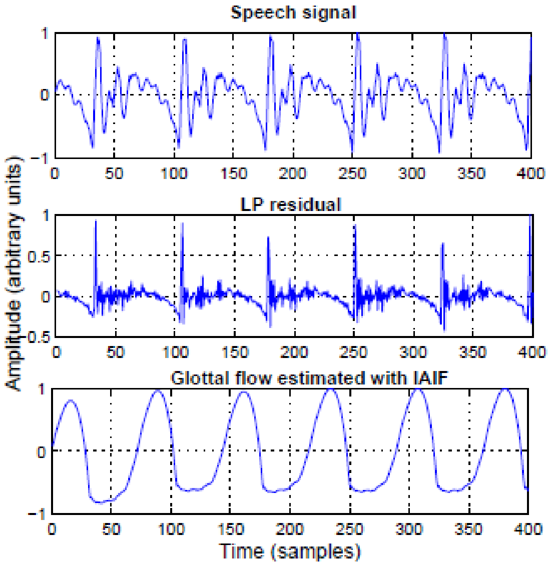


Figure 7: Extracting voice source features from a voice signal (Reylonds and Douglas, 2002)

5- High- level features

High level features are used to identify the speaker by extracting the patterns of words the speaker is usually using and repeating. The utterance of the speaker is changed into tokens, the sequence of repeating these tokens are used to discriminate between different speakers.

High-level features focus on the conversation level of the speakers, they describe the speaker characteristic using the vocabulary that the speaker tends to use in his conversations like “OK”, “Oh yeah” and others… these words are the keys for extracting high-level conversational features.

2.7 Features Extraction methods

There are many methods that are used in extracting features from speech signals, the most famous methods used in this domain are: 1) Mel Frequency Cepstral Coefficients (MFCC) and 2) Linear Predictive Coding (LPC), in this section a brief about each of these methods will be introduced.

2.7.1 Mel Frequency Cepstral Coefficients (MFCC)

MFCC is considered one the most famous methods used in feature extraction (Namrata, 2013), it relies on the spectral form (Shrawankar and Thakare, 2013), therefore it is considered sensitive to noise, this is the main problem in using MFCC. However, (Tyagi, et al., 2005) proposed an important modification for MFCC algorithm; these improvements solved the noise sensitivity problem in MFCC. The following steps describe MFCC method (Namrata, 2013):

- **Pre-emphasis**

  The voice signal is normalized in order to make it less sensitive to finite precision influence.

- **Voice signal framing**

  The voice signal is divided in this stage into *N* number of short duration frames, the duration of these frames is between 20-30 millisecond with overlapping of about 20 milliseconds; 10 milliseconds in the beginning of the frame and 10 milliseconds at the end of the frame.

- **Windowing the signal**

After finishing the framing stage, the framed signal is multiplied by a window function that will smoothen the signal in order to perform the Discrete Fourier Transform computation (DFT) (Winograd, 1978), the main goal of using DFT is to make the beginning and the end of the voice signal goes to zero so that any unwanted artifacts in the windowed signal will be removed (Chang, 2015). DFT assumes that the voice signal keeps repeating continuously until it is broken and discontinued, then it will be processed by the window function as described before. Figure (8) and figure (9) illustrate the windowing process for a discontinued voice signal.

Figure 8: Windowing process for a discontinued voice signal

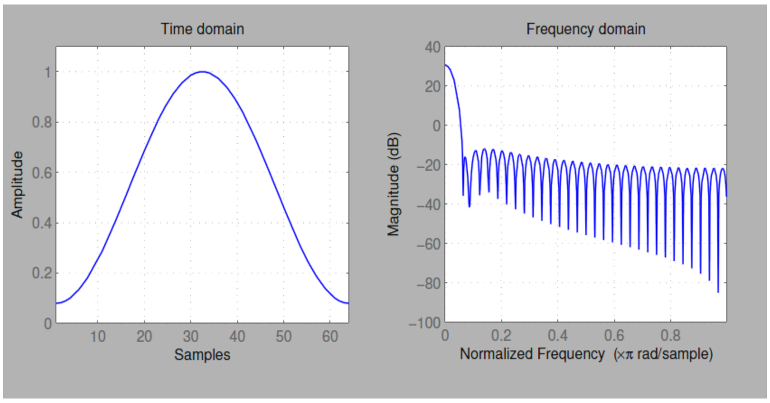
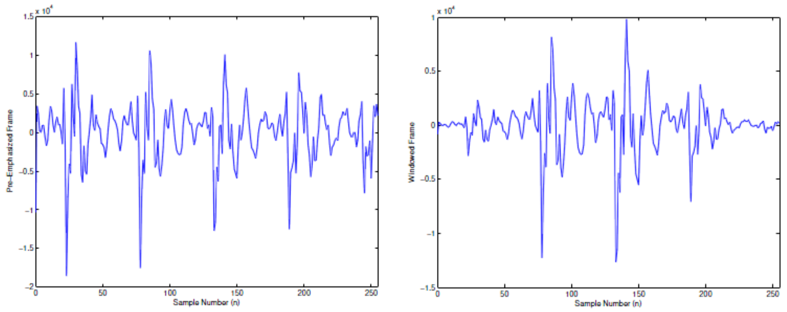


Figure 9: Windowing process for a discontinued voice signal



Pre-Emphasized Frame of audio $N = 256$ Windowed Frame $N = 256$

- **Fast Fourier Transformation (FFT)**

  After completing the windowing step, frequency components will be extracted after deriving FFT for each frame.

- **Mel scale filter bank**

After studying the human ear, scientists found that an ear acts like a bunch of filters that can process certain frequency components (Shrawankar and Thakare, 2013). From here, it was found that the hearing system in humans can be partially modeled using a set of band filters, these filters can only cover the Mel-frequency scale, and because the relationship between Mel-frequency scale and the frequency scale is non-linear, then these filters will not be equally distributed along the scale, actually there will be more filters in the low frequency regions and less filters will assigned on the high frequency regions. Figure (10) shows 24-band Mel frequency filter banks, while figure (11) shows the power spectrum through the 24-band Mel-frequency filter banks.

Figure 11: 24-band Mel frequency filter bank

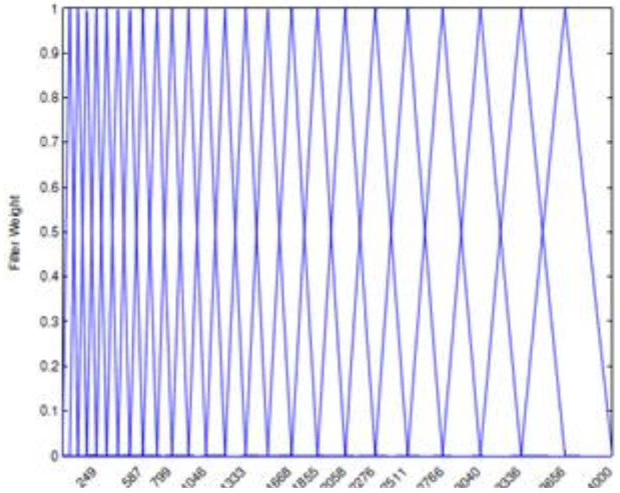
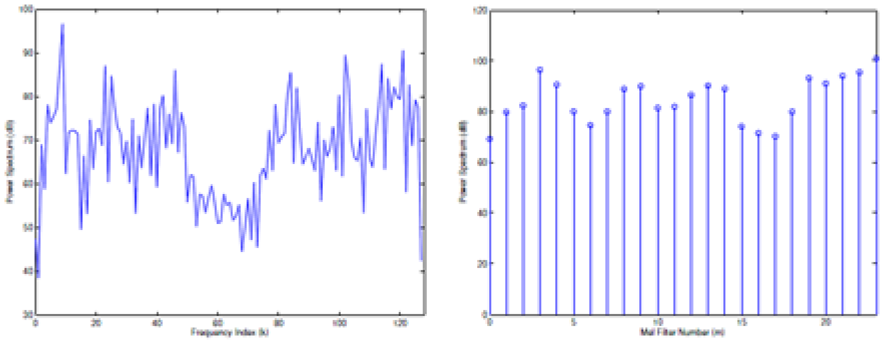


Figure 10: The power spectrum through the 24-band Mel-frequency filter banks



Power Spectrum of the Frame
$N = 256$

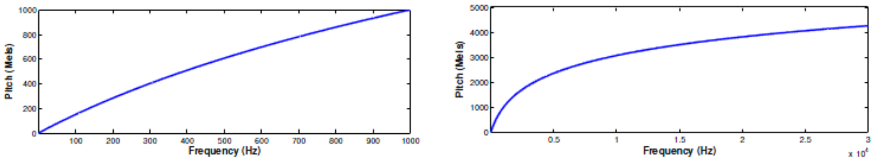
Power Spectrum in the Mel Frequency Domain
$N = 256, M = 24$

- **Log**

  After completing the previous step, the logarithm of the Mel scale filter bank output is calculated.

  Based on experiments; Mel-frequency scale is nearly linear in the period between 1- 1000 Hz, and it changes for the higher frequency to become like the logarithmic plot. Figure (12) shows the plot of pitch (Mel) in different frequency scales.

Figure 12 : The plot of pitch (Mel) in different frequency scales

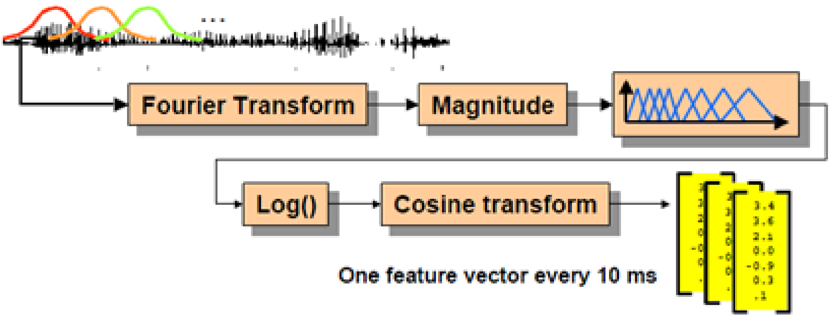


- **Extracting MFCC features**

  The MFCC features are identified by calculating the log of the outputs of a Mel- frequency filter bank. And then applying the Discrete Cosine Transform (DCT) (Nasir et al., 1974). The final MFCC feature vectors are obtained by retaining about 12-15 lowest DCT coefficients.

Figure (13) shows MFCC algorithm steps:

Figure 13: MFCC algorithm steps (Reylonds and Douglas, 2002)



2.7.2 Linear Predictive Cepstral Coefficients (LPCC)

Linear Prediction Coding (LPC), also known as all-pole model or the Autoregressive (AR) model, is another method for spectral envelope estimation. LPC is known by its sufficient interpretation in both time domain and frequency domain. LPC main task is predicting the future voice samples by linearly linking the past voice samples (Shrawankar and Thakare, 2013). The main steps of LPC are as follows (Mohammed et al., 2013):

- **Pre-emphasis**
  The voice signal is normalized in order to make it less sensitive to finite precision influence.
- **Frame blocking**
  The normalized signals is divided into typically 20ms frames (Shrawankar and Thakare, 2013).
- **Windowing the signal**
  After finishing the framing stage, the framed signal is multiplied by a window function which will smoothen the signal, and decreases gaps between the beginning and the end of the frames.
- **Autocorrelation**
  After completing the previous step, the frames are then autocorrelated.
- **LPC analysis**
  The values generated from the previous step are processed using the Levinson-Durbin recursion method, and the output is LPC parameter set.
- **LPCC extraction**
  After computing the LPC parameter set, the LPCC will be generated.

MFCC is based on a perceptual frequency scale (Mel-frequency scale), LPCC does not depend on such a scale, this opened the idea of developing a new analysis principle which is Perceptual Linear Predictive (PLP) analysis (Chang, 2015).

2.8 Speaker Modeling

A speaker model is created using the feature vectors generated from the previous steps; this is accomplished in the enrollment phase. The following machine learning techniques are usually used to build a speaker's model (Chang, 2015).

- **Hidden Markov Models (HMM)**

In text independent speaker recognition only single state HMM is needed to represent the speaker, this type of modelling called Gaussian Mixture Models. While in text dependent systems, words and phrases need multi-state HMMs to be modeled properly (Chang, 2015).

- **Neural Networks (NN)**

The most important benefit of using neural networks in speaker recognition systems is that a single neural network can be used for two purposes at the same time, namely, feature extraction and speaker modeling. The use of a single neural network in this way will allow optimization for both feature extraction and speaker modeling at the same time (Chang, 2015).

- **Support Vector Machine (SVM)**

The Support Vector Machine (SVM) is a binary classifier that creates a boundary between two different classes. SVM is used in speaker recognition by modelling the boundary between two classes, one is the target-speaker feature's vector labeled as (+1) in figure (14), the other represents the training feature from the speaker that should be recognized or

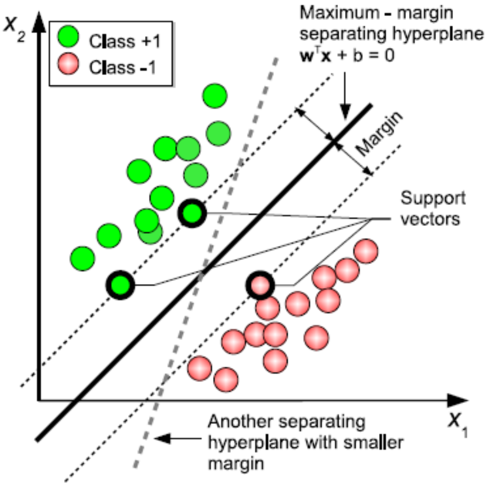

X_2
Class +1
Class -1
Maximum - margin separating hyperplane $w^T x + b = 0$
Margin
Support vectors
Another separating hyperplane with smaller margin
X_1

Figure 14: Support Vector Machine (SVM) Adapted from (Reylonds and Douglas, 2002)

the imposter which is labeled as (-1) in figure (14). Using the two features described before, SVM determines the decision boundary that maximizes the margin space between the two classes which will result of deciding whether the model understudy is similar to the stored model or not (Chang, 2015).

2.9 Related Work

In this section we will summarize some of the successful applications and researches that studied and implemented speaker recognition systems in the last few years.

(Stajano, 2011) developed a system that uses tokens instead of passwords in the authentication process, this system was called Pico. Later, (Toader and Stajano, 2014) improved the capabilities of Pico by adding more unlocking procedures to the designated authentication system. Picosiblings are small tools that are physically connected or embedded in everyday used items like watches, necklaces, bracelets and other jewelry. Pico system unlocks when it is successfully communicates with Picosiblings. Picosiblings send secret sequences to Pico, when all the secrets are completely collected, Pico unlocks and then it can be used. (Toader and Stajano, 2014) clearly indicated in his research that the main weak point in Pico is that anyone who has the required Picosibling can unlock Pico and be authenticated. This is because there is no way to verify the real identity of the user, and no tool to determine whether this user is allowed to unlock Pico or not.

(Toader and Stajano, 2014) proposed a solution for this problem in his research, his solution was to include more levels of security to the authentication system of Pico, these levels included biometrics and behavior metrics, like voice recognition, face recognition and Iris recognition (Toader and Stajano, 2014). Picosiblings functions were later moved to mobile devices, because the new Picosiblings application runs on Android devices. This application gives specified weights to every metric, these weights are determined depending on the degree of accuracy this metric can achieve in identifying the actual user. Each metric estimates the probability that the actual user is using the device, the application then calculates the weighted sum of the generated probabilities; the result summation is considered as the overall level of confidence, Pico uses this level of confidence to unlock all the applications that are assigned a lower level of confidence than the generated one. Pico uses Recognito library for using voice recognition in its application (Anon, 2015).

(Sánchez, 2010) used C++ to develop a speaker recognition tool for laptops; he performed the training phase using one word and he used the same word for the testing phase. Depending on Euclidean distance, the tool compares the tested speaker model with the stored models; the best match for the tested speaker is the model which has the shortest distance between the tested and the stored models. The main issue here is that the distance between the tested model and all the stored models should be calculated in order to identify the speaker's identity.

University of Avignon developed Alize using C++ (Anon, 2015), it is an open source platform that is used for both voice and image recognition.

Alize has many configuration options, these options help in generating more accurate results.

University of Erlangen-Nuremberg developed The Java Speech Toolkit (JSTK) (GitHub, 2015), this toolkit is used for several voice related tasks like voice and speech recognition. Using Java to build this toolkit makes it easier to implement JSTK on Android devices. Also this allowed the device to locally perform some of the calculations related to voice tasks rather than sending them to the associated server (GitHub, 2015).

Carnegie Mellon University also has its own contribution in the voice and speech recognition domain; researchers in CMU developed Carnegie Mellon University's (CMU) Sphinx, which was mainly developed to perform speech recognition tasks. Later (CMU) Sphinx's functionality was extended to perform speaker recognition tasks (Cmusphinx.sourceforge.net, 2015).

Python is widely used in developing biometrics related systems, in the speaker recognition area; python is used in several well-known applications like (SPEAR) (Pypi.python.org, 2015) and voiceid (Code.google.com, 2015), also it is the main programming language used for developing our application.

Processes like feature extraction/ reduction and models building, are all basic processes in almost all of the speaker recognition systems. These processes are computationally expensive; therefore some researchers proposed server based speaker recognition approaches. These approaches used mainly distributed architectures that contain frontend and backend. Noise removal and features extraction are performed in the frontend while matching models and testing are performed in the backend. (Chowdhury et al., 2010) study is an example for the server based speaker recognition systems, Gaussian Mixture Model- Universal Background Model (GMM-UBM) models are used in this approach. Similarly, (Li et al.,2004) used in his approach server-based speaker recognition system using (GMM-UBM).

Using voice samples from different sources of databases, (Brunt et al.,2013) proposed a speaker recognition system that runs entirely on an Android device. The proposed system extracts the voice features and represents them as a distance vectors, then the system calculates the Euclidean distance between the stored samples and the test samples, depending on this calculation it identifies the speaker.

Chapter 3

Methodology

This chapter will describe the main modules used to build the proposed speaker recognition system; the modules are 1) The Preprocessing Module for data preparation, 2) Feature Extraction Module, and 3) Recognition Module.

3.1 Introduction

This study aims to build a speaker recognition system, and then evaluate its performance in recognizing the speaker's identity. The proposed system implemented a various number of features extraction techniques like MFCC and LPCC. Also the system tested the performance of well-known recognition methods like Universal Background Model (UBM), Gaussian Mixture Model (GMM), Continuous Restricted Boltzmann Machine (CRBM), and Joint Factor Analysis (JFA).

All the mentioned methods performance will be examined and investigated using several values of their algorithms parameters.

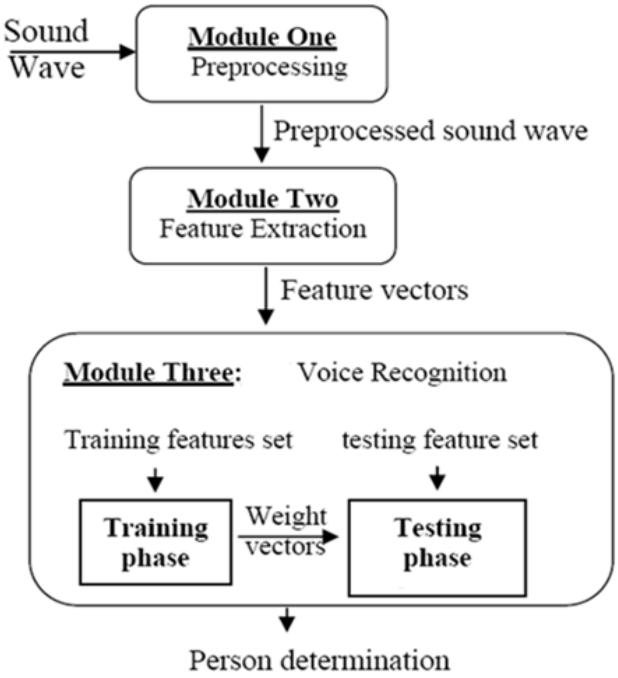
3.2 The Proposed Model

The main functionality of any speaker recognition system is to determine the specific features of the speaker's voice sample and then match them with the stored models to specify the speaker's identity. The proposed speaker recognition system contains three main modules as shown in figure (15) (El Hannani et al., 2009).

The recognition system is initiated by recording the speaker's utterance, then the recorded samples will be preprocessed. The output of the preprocessing phase will be passed to the feature extraction module to extract the features which will represent the features vectors dataset.

At the end of the previous module, the extracted features will be passed to the recognition module, which consist of two basic phases, the training phase (enrollment phase) and the testing phase (recognition phase). Finally the trained system will be able to identify the speaker's identity.

Figure 15: The Developed system flow control



3.2.1 Preprocessing Module

In order to start the preprocessing phase, the voice samples should be recorded by different speakers using a microphone. The microphone system is responsible on converting the voice sample into an electronic signal; this signal will be stored in the system. The recorded samples used in the proposed system has the following characteristics:

- **Sampling rate:** 8 KHz
- **Bit per sample:** 16 bit
- **Number of channels:** mono or stereo.

Many voice recognition applications use mono channel rather than stereo channel; because mono channel needs less time to perform the recognition process, also human voice usually has a low bandwidth between 100 Hz and 8 kHz. Processing 8000 samples/sec is much computationally inexpensive than processing 16000 samples/sec. however, some researchers prefer 16 kHz signals because It can provide more accurate information than 8 kHz signals.

The inputs for the preprocessing phase are the recorded voice files, these files will pass through the following processes:

- **Voice Activity Detection (VAD):**

  After recording the speaker's voice, the captured signal must be filtered using the Voice Activity Detection (VAD) algorithm. The purpose of VAD is to get rid of the silence part of the captured signal (Ramírez et al., 2004), if VAD is not applied on the voice signal then the training of the model will be extremely biased. In our experiment the voice signals used are mostly have no vocal noise, therefore the criteria used to delete the silence part of the voice signal is a simple energy based method. This method removes the frames that have an average energy of less than 0.01 times the average energy of the whole recoding. It was found that the energy-based approach is applicable on performing VAD on a database of signals, but for real time speaker recognition, which uses Graphical User Interface to capture the utterances; the energy-based approach is slow and not practical. However in our experiment we tried another techniques, we used Long-Term Spectral Divergence (LTSD) for signal silence frames removal (Ramírez et al., 2004), in addition to a noise reduction technique from SOX (Wikipedia, 2015), SOX is an open source tool which we included in our proposed system (Sox.sourceforge.net, 2015).

  LTSD detects silence frames in the voice signals by applying the following steps:

  1. LTSD splits the utterance into overlapped frames.
  2. It gives a score for each frame by calculating the probability that there is a voice activity in that frame.

3. The probability value will be accumulated, so that it will be able to detect the voice activity in the whole interval of the signal.

Figure (16) displays the LTSD algorithm.

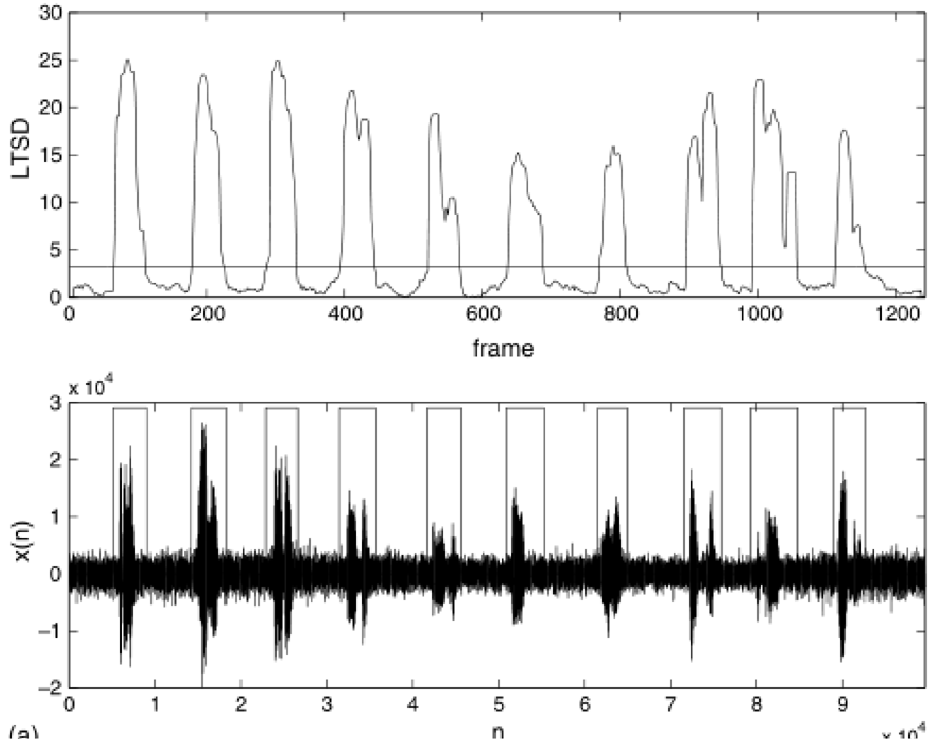


Figure 16: LTSD algorithm

3.2.2 Feature Extraction Module

The second step in our proposed system is features extraction from the sound signal. The extracted features are the core components for the speaker recognition step in our system. There are many known methods for sound features extraction, in our proposed system we used the following Feature extraction methods after applying Discrete Fourier Transform (DFT) on the sound signal under testing.

1. **Mel-Frequency Cepstral Coefficient:**

As described previously in chapter 2, MFCC is a valid representation of the short-term spectral features of a voice signal; it is a linear cosine shape of a log power spectrum of a nonlinear mel-scale frequency (Xu, et al. 2005) (Wikipedia, 2015). In general, MFCC features extraction starts by splitting the voice signal into equal short-term frames with length of *L*, with an overlap distance between the neighboring frames equal to *R*. the generated frames are then windowed as shown in Figure (18). In figure (17) the process of extracting MFCC features is explained.

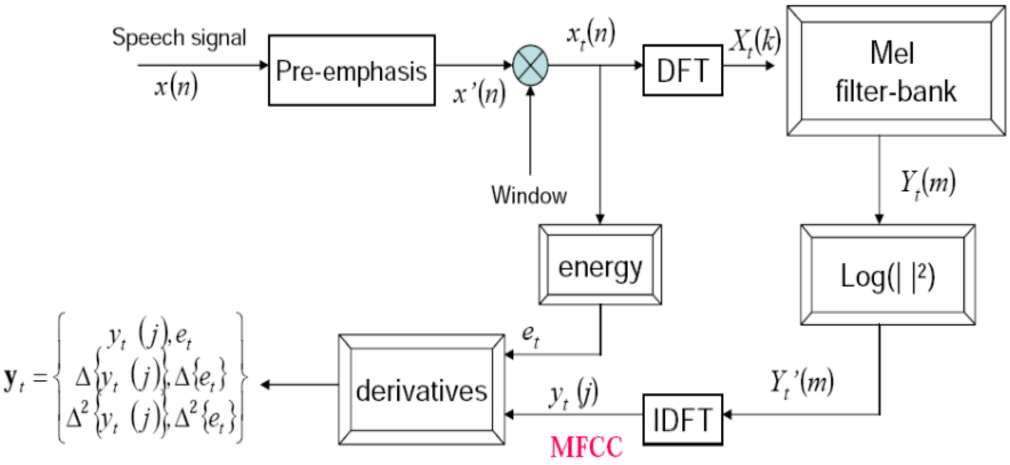

Speech signal
$x(n)$
Pre-emphasis
$x'(n)$
Window
$x_t(n)$
DFT
$X_t(k)$
Mel filter-bank
$Y_t(m)$
Log(| |²)
$Y_t'(m)$
IDFT
MFCC
$y_t(j)$
energy
e_t
derivatives
$\mathbf{y}_t = \left\{ \begin{matrix} y_t(j), e_t \\ \Delta\{y_t(j)\}, \Delta\{e_t\} \\ \Delta^2\{y_t(j)\}, \Delta^2\{e_t\} \end{matrix} \right\}$

Figure 17: MFCC feature extraction process

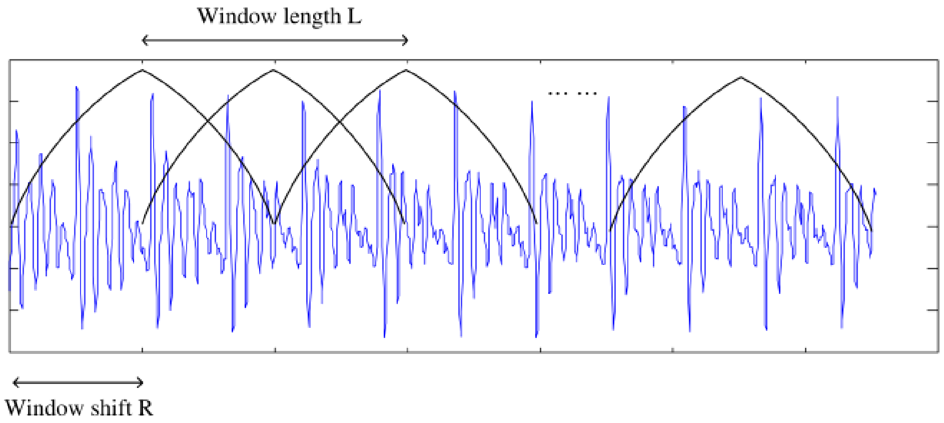


Figure 18: Windowing frames

After windowing the short-term frames, the spectrums of the windowed signals must be computed, this is achieved by applying Discrete Fourier Transform (DFT) on the windowed signals. DFT gives a complex number *X*[*k*] that represents the magnitude and the frequency level for *N* discrete frequency bands in the original signal. After this, MFCC wrapping is applied on the signals spectrums, and it is calculated using the following equation (Feng, 2015):

$$M(f) = 2595 \ \log_{10}\left(1 + \frac{f}{700}\right)$$

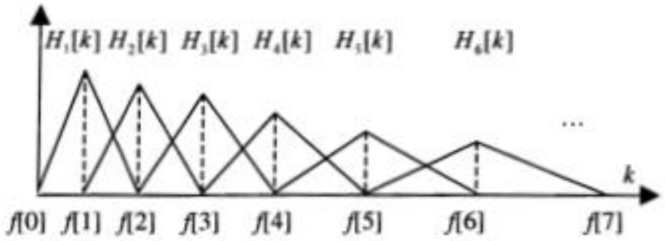


Figure 19: Filter Banks (6 filters) adapted from (Chang, 2015)

After MFCC wrapping is completed, we apply the filter banks on the Mel-scale spectrum of signals, then the logarithm of energy for each bank is calculated using the following equation (Ee.columbia.edu 2015):

$$E_i[m] = \ \log\left(\sum_{k=0}^{N-1} X_i\,[k]^2 H_m[k]\right)$$

Then we calculate the Discrete Cosine Transform (DCT) of $E_i[m]$ to get an array C_i ,where C_i is equal to (Ee.columbia.edu 2015):

$$c_i\,[n] = \sum_{m=0}^{M-1} E_i\,[m]\cos\left(\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right)$$

After this we use the first *k* terms in the generated array C_i which represents the extracted features form the voice signal. These features will be used in the training stage of the proposed application. The number of the first *k* terms to be used in the training stage is different from case to another.

2. Linear Predictive Coding (LPC):

Linear Predictive Coding (LPC) main principle is that for a short time a nonlinear signal becomes a linear combination of short previous signals.
The signal *s[n]* is predicted by a linear combination of its previous values.

The prediction equation is defined as follows (Chang, 2015):

$$\tilde{s}[n] = \sum_{k=1}^{p} a_k s[n-k]$$

Where $s[n]$ refers to the signal, a_k are the prediction coefficients and $\tilde{s}[n]$ is the predicted signal. The prediction error signal, is computed as follows (Chang, 2015):

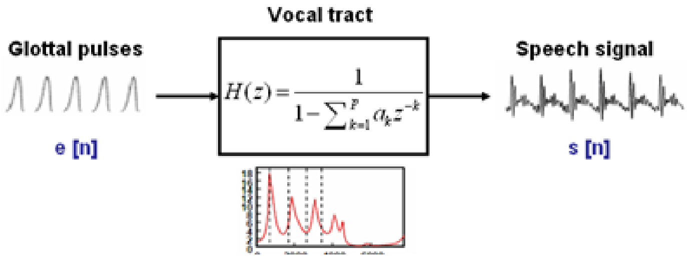
$$e[n] = s[n] - \tilde{s}[n]$$

The coefficients a_k values are specified by minimizing the error signal's energy $E[(e[n])^2$ using the Levinson-Durbin algorithm (Ee.columbia.edu 2015).

Figure (20) shows, $s[n]$ is the speech signal, $e[n]$ is the voice source (glottal pulses), and $H(z)$ is the response of the vocal tract filter (Martínez-Romo et al., 2012).

The following equation describes figure (20) (Martínez-Romo et al., 2012):

Figure 20: Linear Predictive Coding Adapted from (Chang, 2015)



$$e[n] = s[n] - \tilde{s}[n]$$

$$= s[n] - \sum_{k=1}^{p} a_k s[n-k]$$

$$E(z) = S(z)[1 - \sum_{k=1}^{p} a_k z^{-k}]$$

$$H(z) = \frac{S(z)}{E(z)}$$

From the derived equation we can conclude the spectral model that demonstrates the vocal tract as follows (Chang, 2015).

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}$$

3.2.3 Recognition Module

After processing the voice signal in the preprocessing module, and pass it to the features extractor module the speaker recognition module is called. The recognition process is performed using several methods each one of them is applied in a separate implementation in order to study and analyze the performance of these methods. Preparing a reliable scientific comparison between these methods is one of the important goals in this thesis.

The modelling methods tested in our system are mainly Gaussian Mixture Model (GMM), Universal Background Model (UBM), Continuous Restricted Boltzmann Machine (CRBM), Joint Factor Analysis (JFA). The classification process is divided into two phases: training phase and testing phase, figure (21) and figure (22) describe the recognition module main processes.

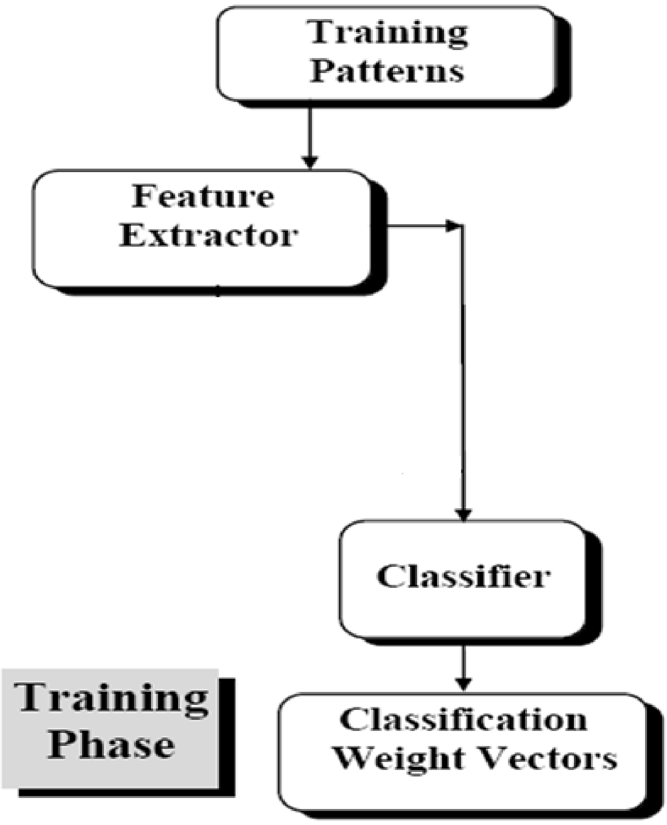


Figure 21 : Training Phase

- **Training phase**

  The classifier in the system is trained on a set of the extracted features from the voice signal. The training process core is to generate and assign weights to the vectors that represent the features coefficients.

- **Testing phase**

  The trained classifier from the training phase uses the weighted vectors generated from the training phase. It assigns the unknown input pattern to one of the classes based on the extracted feature vectors. Once the tested module achieves the maximum likelihood percentage in relative to a stored model, then the recognition process declares the found identity of the current speaker.

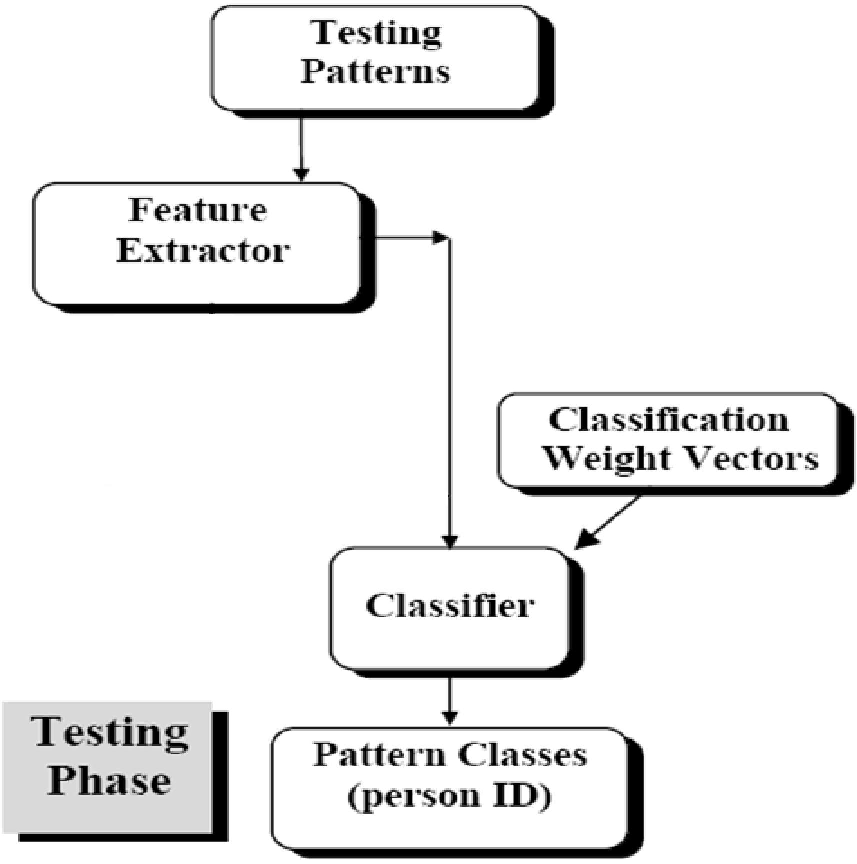


Figure 22: Testing Phase

3.2.3.1 Modelling techniques used in the recognition module.

- **Gaussian Mixture Model (GMM):**

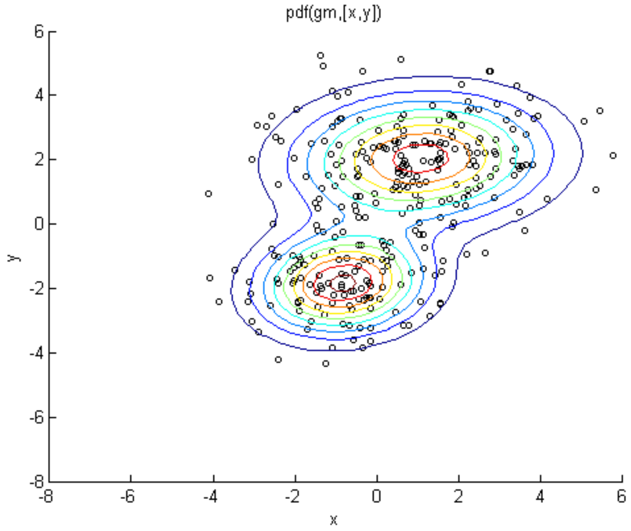
GMM is a model that describes the various distributions of all the feature vectors that are extracted from a voice signal; it is used frequently in acoustic learning like speaker recognition task.
GMM calculates the probability of whether a feature vector belongs to a specific model or not using the following (Reynolds and Rose, 1995):

$$p(x|w_i, \mu_i, \Sigma_i) = \sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

$$\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right)$$

$$\sum_{i=1}^{K} w_i = 1$$

Figure 23: A Two-Dimensional GMM with Two Components



By analyzing the previous calculation, it is clear that GMM is a combination of a Multivariate Gaussian Distributions, which assumes that feature vectors are independent. Because of this assumption, the diagonal covariances are used in the proposed application. Figure (23) shows how GMM describes the distribution of feature vectors using multiple clusters.

The training phase of GMM is performed by finding the best parameters forμ_i, $\sum_i$, w_i, so that the model fits all the training inputs with maximized likelihood. The maximized likelihood is achieved using the Expectation Maximization algorithm (Bilmes, 1998).

One iteration of the expectation maximization algorithm in GMM model is as follows:

1- E- Step for feature vector which is represented by data points: the estimation of the probability that each Gaussian generated specific data points is calculated using the following equation (Reynolds and Rose, 1995).

$$p(x|w_i, \mu_i, \Sigma_i) = \sum_{i=1}^{K} w_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

2- M- Step, in this step the main goal is to maximize the likelihood factor and this can be done by modifying the values of the GMM parameters. In this step a new hidden variable is used to specify that *i*-th data point is generated by *j*-th Gaussian. In this step we can use the maximization of the expectation of the log of likelihood rather than the maximization of likelihood of data with respect to the same parameter which is *Z* in this following equation (Bilmes, 1998):

$$\mathrm{Q}(\theta', \theta) = \mathrm{E_Z}\langle \log \mathrm{p}(\mathrm{X}, \mathrm{Z}) \,|\theta\rangle$$

Where θ is the current parameter, and θ' are the estimated parameter for the next iteration. Using the constraint $\sum_{i=1}^{k} w_i = 1$ and applying Lagtange multiplier we get the following equations (Bilmes, 1998):

$$J(\theta', \theta) = Q(\theta', \theta) - \lambda \left(\sum_{i=1}^{K} w_i - 1 \right)$$

Set derivatives to zero, we can get the update equation

$$Pr(i|x_j) = \frac{w_i \mathcal{N}(x_j|\mu_i', \Sigma_i')}{\sum_{k=1}^{K} w_k \mathcal{N}(x_j|\mu_k' \Sigma_k')}$$

$$n_i = \sum_{j=1}^{N} Pr(i|x_j)$$

$$\mu_i = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_j) x_j$$

$$\Sigma_i = \left(\frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_j) diag(x_j x_j^T) \right) - diag(\mu_i' \mu_i'^T)$$

$$w_i = \frac{n_i}{N}$$

After completing the training stage, the GMM model will be able to determine the feature vectors for every data point that belongs to a specific model. In the speaker recognition process, GMM will be trained for every speaker, then the input signal for the speaker understudy will be analyzed and a list of its extracted features will be generated. After this the overall likelihood will be computed for all the feature vectors to identify to which model these vectors belong, then the model which fits best the input signal will be chosen as the answer for the question “who is speaking now?”.

A new GMM model:

In the proposed system, a simple enhancement was performed on the original GMM method. GMM uses a random initialization of the means of all the components in the training stage. But in the proposed system, K-means is used to cluster all the features vectors in separate clusters (Hartigan and Manchek,1979) (Doc.madlib.net 2015), then we used the central points for the cluster to initiate the training of GMM. By using the central points for initiating the training process in GMM, the training process will speed up and the training results will be enhanced also, this is tested and proved in the performance part of this research. Additionally, two K-means algorithms were experimented in this enhancement; K-mean ++ (Arthur and Vassilvitskii, 2007) and K-mean II (Bahmani et al., 2012), and it is found that K-mean II gives more accurate results than K-mean ++, so it was applied in the enhanced GMM.

- **Universal Background Model (UBM)**

Universal Background Model (UBM) is a type of GMMs that is trained on a huge number of models (speakers), it models the most common features of the human voices. UBM is used as an imposter in the proposed system using the formula in (Reynolds, Quatieri and Dunn, 2000) and likelihood ratio is calculated in order to generate decisions as described in (Reynolds, Quatieri and Dunn, 2000), in our experiment we used GMM models for users that is described from the pre-trained UBM as mentioned in (Reynolds, Quatieri and Dunn, 2000)

- **Continuous Restricted Boltzmann Machine (CRBM)**

Restricted Boltzmann Machine (RBM) is a type of neural networks that is able to learn a probability distribution using its binary inputs; RBM is generative and contains two layers of neural network. CRBM processes a stream of continuous real time input data. RBM main benefit is its ability to generate hidden layers similar to the input visible layer. RBMs are widely used as a modelling tool in voice recognition domain. Figure (24) and figure (25) show original MFCC data and the output of regenerated data from CRBM.

CRBM if used in a speaker recognition system instead of GMM as a modelling tool; then a CRBM model for every speaker must be trained. By estimating the error on the reconstruction error in CRBM, the recognized speaker will be the one with the CRBM which produced the lowest reconstruction error.

Figure (24) shows the first three dimensions of the same woman's MFCC feature reconstruction by a CRBM with 50 neuron hidden layers, while figure (25) shows the first three dimensions of a woman's MFCC features.

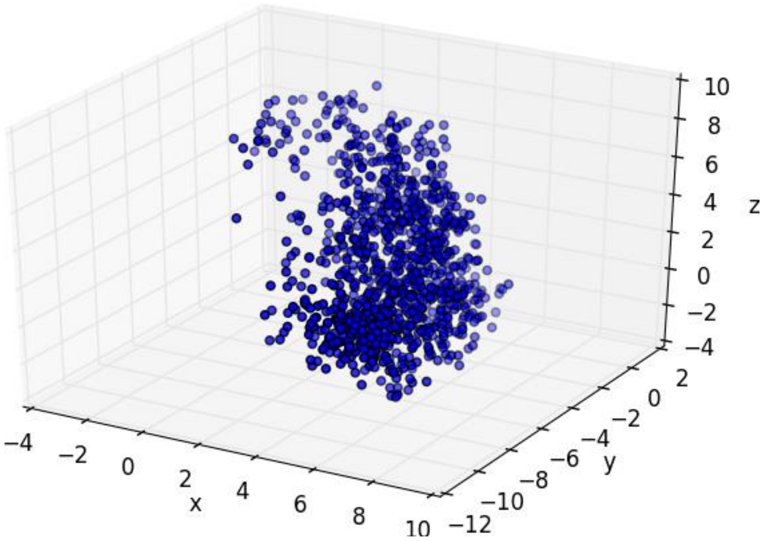


Figure 24: The first three dimension of a woman's MFCC feature

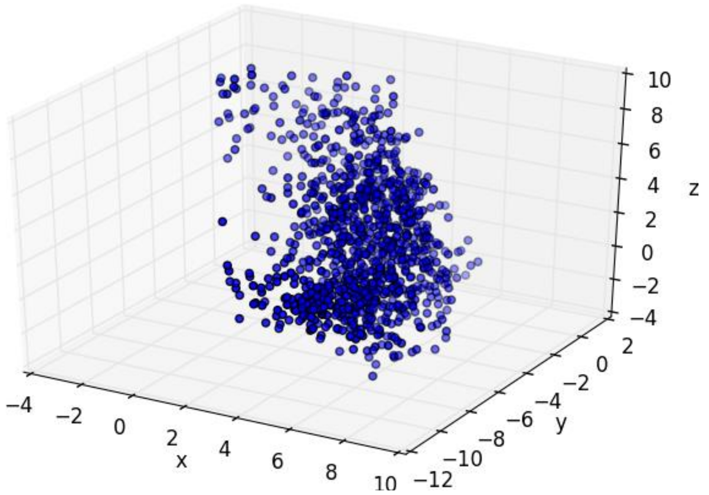


Figure 25: The first three dimension of the same woman's MFCC feature regenerated by a CRBM with 50-neuron hidden layer.

- **Joint Factor Analysis (JFA)**

Factor Analysis is a classification method that is known by its flexibility in processing various types of data points in a training set. Joint factor analysis proved by experiments that is it the best factor analysis method to be used in voice recognition (Kenny, 2005) (Kenny et al., 2007).

JFA represents the speaker by a supervector, for example $C \times F$ dimension vector, C represents the number of components in UBM which are trained by GMM on all the training data. And F is the dimension of the acoustic feature vector. In order to get the supervector for an utterance the entire C means vectors must be concatenated in the trained GMM model. JFA describes the supervector using the following equation (Kinnunen and Haizhou, 2010) :

$$\vec{M} = \vec{m} + vy + dz + ux$$

$\vec{m}$ represents a supervector that is trained by UBM, v represents $CF \times R_s$ dimension matrix, u represents $CF \times R_c$ dimension matrix, d represents a diagonal matrix. All these variables are considered independent and they retain the same values after the training. The rest of variables x, y, z are matrices that are calculated for each utterance. In this formula, $\vec{m} + vy + dz$ represesnts Inter-speaker variablilty, and ux represents Inter-channel variability. R_c stands for channel rank, R_s represents speaker rank. The training of JFA is to calculate the best values for v, u, and d to fit most of the training data.

Chapter 4

Implementation

4.1 Introduction

The speaker recognition system was developed using the following programming languages: Python, Matlab and C++. However, it was mainly developed using python. The proposed application relies on two important Python libraries, numphy and scipy libraries (Numpy.org, 2015) (Scipy.org, 2015).

All the methods that were described in chapter (3) are programmed and implemented in our system; in this section we will give a brief about the most important methods that is considered vital for the methodology of the proposed system.

4.2 Preprocessing Module

4.2.1 Voice Activity Detection (VAD):

Three types of VAD filters were developed and used in the system; they are located in the folder **src/filters**, these filters are as follows:

- **silence.py**, this filter uses energy based VAD algorithm.
- **Lstd.py** is a wrapper for LTSD algorithm; it uses the **Pyssp** library (Pypi.python.org, 2015).
- **Noisered.py** is a wrapper for SOX noise reduction tool; it works depending SOX (Sox.sourceforge.net, 2015) which was installed in the system.

4.3 Features extraction Module

Features extractor tools are located in **src/feature** folder; it contains the following features extraction methods:

- **MFCC.py** is a self-implemented MFCC feature extractor.

- **BOB.py** is a wrapper for the MFCC feature extraction; it uses the BOB library (Anjos et al., 2012).
- **LPC.py** is a LPC feature extraction tool; it uses scikitstalkbox to work properly (Scikits.appspot.com, 2015).

After conducting many times of experiments using the test script located in **src/test/test-feature.py,** it was found that the best values of the parameters used in the feature extraction tools are as follows:

a) The following are common parameters used in the feature extractors used in the system:
Frame size: 32 ms.
Frame shift: 16 ms.
Pre-emphasis coefficients: 0.95.

b) MFCC parameters:
Number of ceptral coefficients: 15
Number of filer banks: 55
Maximal frequency of the filter bank: 6000

c) LPC Parameters:
Number of coefficients: 23

4.4 Recognition Module

4.4.1 GMM

When we tested GMM, we tried GMM that is embedded inside scikit-learn (Pedregosa et al., 2011), and we tried GMM from Pypr (Pypr.sourceforge.net, 2015), after conducting several trials on these two types of GMM we found that both of the two implementations of GMM have a performance problem, they both showed that they are time consuming in comparison to our new developed GMM.

In the proposed GMM the initiation inputs were chosen by applying K-mean II algorithm, and the concurrency support was implemented and located in **src/gmm** folder. The new GMM was developed using C++ and needs G++>= 4-7 to be compiled. Additionally, the proposed GMM has Python bindings which mean that it is fit to be used within a python package. The new GMM showed a remarkable enhancement in speed and accuracy .

In the proposed system GMM was used with 32 components, this number of components was specified after conducting many trails. The covariance matrix of every Gaussian component is assumed to be diagonal because each dimension of the feature vector is independent.

4.4.2 JFA (Joint Factor Analysis)

After studying the original algorithm of JFA in (Kenny, 2005), we found that there is a simplified version of this algorithm is presented by (Kenny et al., 2008) to train the JFA model. The JFA implementation is induced from ([http://speech.fit.vutbr.cz], 2015). Generating JFA features file is completed using the file **test/gen-features-file.py**. When the files **train.lst**, **test.lst** and **enroll.lst** are put in the folder **JFA/feature-data** we can run the file **run_all.m**, this file will perform the training and testing and the file **exp/gen_result.py** will compute the accuracy of the results.

After checking the accuracy of the results of using JFA algorithm, we found that it couldn’t reach the level of accuracy and speed of applying our new GMM and MFCC algorithms on the same features dataset. After conducting several trials we found that JFA model needs more data than GMM, because JFA needs more variation of data which is implemented as more dimensions in Gaussians. Adding more dimensions in applying JFA means that training of JFA model will consume more time than other methods under study; this is because the estimation of *u*, *v* and *d* doesn’t converge quickly. After studying the results of applying JFA in our system, we found that it is not practical to use JFA algorithm as the core of our real time speaker recognition system.

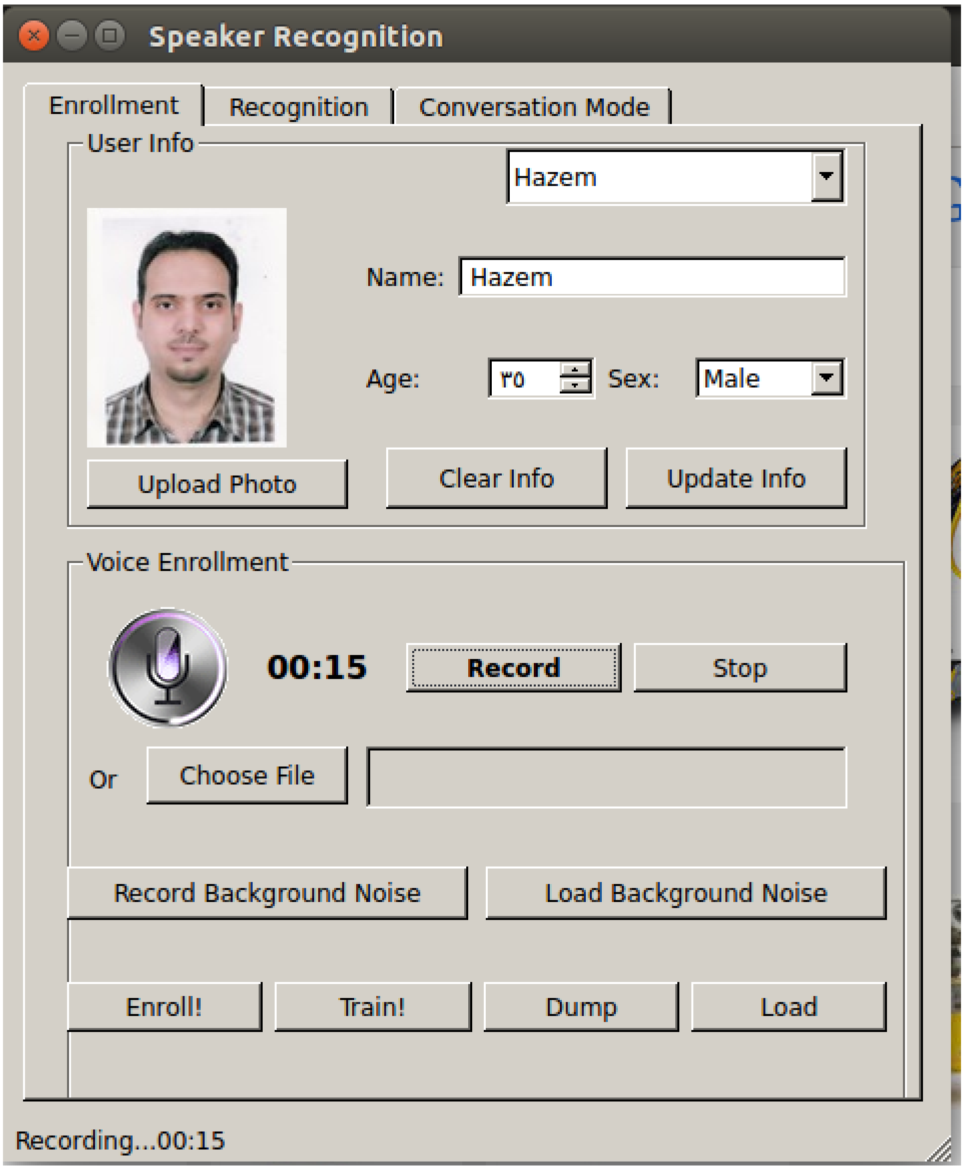
4.4.3 CRBM (Continuous Restricted Boltzmann Machine)

CRBM is implemented using C++, and it is located in **src/nn**. It also has concurrency support.

In the proposed system all the implemented methods and techniques use the same interface and use parameters that can be reconfigured to get the best performance of the speaker recognition system, experiments and testing the parameters are described in the next chapter.

4.5 Graphical User Interface for the system (GUI)

The GUI used in the proposed system is built and designed using PyQt (PyQt, 2015), and the audio functions embedded in this GUI was developed using PyAudio (People.csail.mit.edu, 2015).



Hazem

Name: Hazem

Age: ٣٥ Sex: Male

00:15

Recording...00:15

Figure 26: Enrollment Tab

GUI on the speaker recognition system contains the following tabs:

4.5.1 Enrollment tab

As shown in figure (26), the enrollment tab allows the new user to build a profile that contains some of his personal information like name, age and sex. User also can upload his photo as a part of his identity. If the user already created a profile, he can choose his name from the list of users in this tab and all his information will appear in the personal information fields.

After preparing the profile, the user needs to provide his voice sample in order to train the model. This can be achieved in two ways:

– Enroll by Recording

The user click on record button to start recording his voice, and then he should click stop button to stop and save his voice sample. The recording time is unlimited; it is recommended that the user record his voice for a sufficient period of time in order to feed the enrollment process with enough information for the recognition phase.

– Enroll from Wav Files

User can provide the system with a sample voice by uploading a pre-recorded audio file; the file should be in .wav format. After uploading the wav file the user can train, dump or load this file.

4.5.2 Recognition tab

After providing the system with the required voice sample, the system in this tab will perform the recognition process, and the photo of the user if uploaded will appear, also the name will be shown in the command line. The recognition of multiple users also can be performed in this tab if multiple files are uploaded in the enrollment tab. Figure (27) shows the recognition tab.

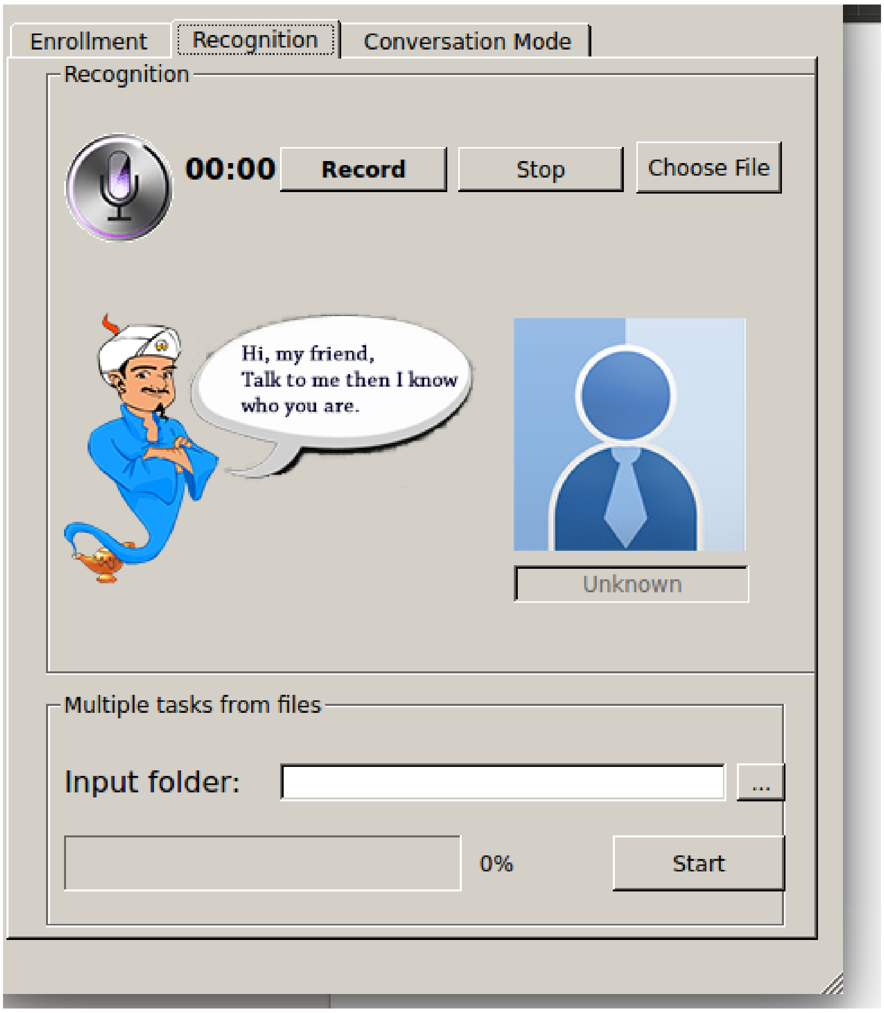


Figure 28: Recognition Tab

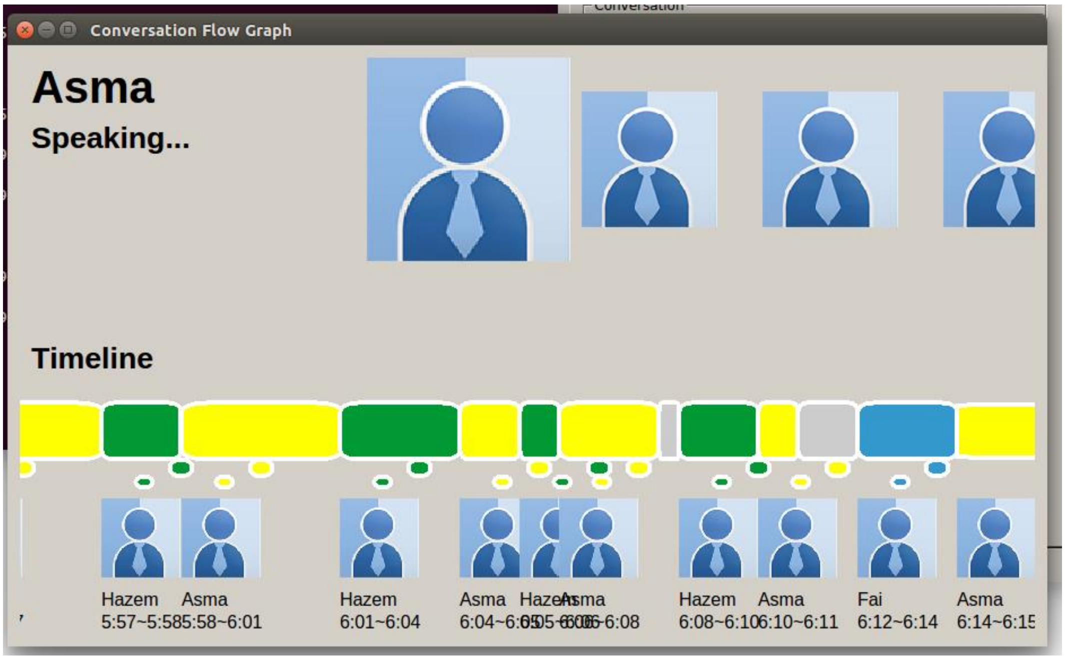


Figure 27: Conversation Mode

- **Conversation Recognition Mode**

In this tab, several speakers can make conversation between each other near the microphone; the system will identify each one of them. The system will continuously collect the voices and recognize each of them nearly immediately.

When the conversation is conducted the system will show the photo and the name of the speaker who is speaking right now. Figure (28) illustrates the conversation mode screen.

4.6 Dataset of sampled human voices

In this thesis, two types of dataset were used, the first one is the extracted features dataset which is counted by thousands, and generated for every speaker model. The second dataset contains samples of human utterance used to test the accuracy of the proposed system, this dataset is acquired from voxforge repository (Repository.voxforge1.org, 2015), this dataset contains many types of human utterance, it contain male voice samples and female voice samples, also it has different types of speaking styles, for example it consists of Reading style, whisper style and spontaneous style.In our thesis we used 102 speakers samples, 60 of them belong to female speakers, and 42 for male speakers. The duration of the selected speeches varies from 5 seconds up to 200 seconds.

Chapter 5

Experiments and Results

5.1 Introduction

This chapter shows various tests performed on the speaker recognition system with the use of the various techniques explained early in this thesis. We implemented the tests using the dataset described in chapter 4, all tests were conducted several times and the speakers used in the training and testing processes were chosen randomly. Additionally, we used different parameters values to study the accuracy degree achieved by the output of the proposed system using these various values.

5.2 Efficiency Test of our GMM

In this test, real MFCC data with 13 dimensions and 20ms frames were used to study the speed and accuracy of the new GMM compared to the scikit-learn GMM version. This test is also performed when training a UBM with 256 mixtures. 10 iterations- with different data size and concurrency- were performed using both types of GMM, and the time consumed when performing these iterations was recorded.

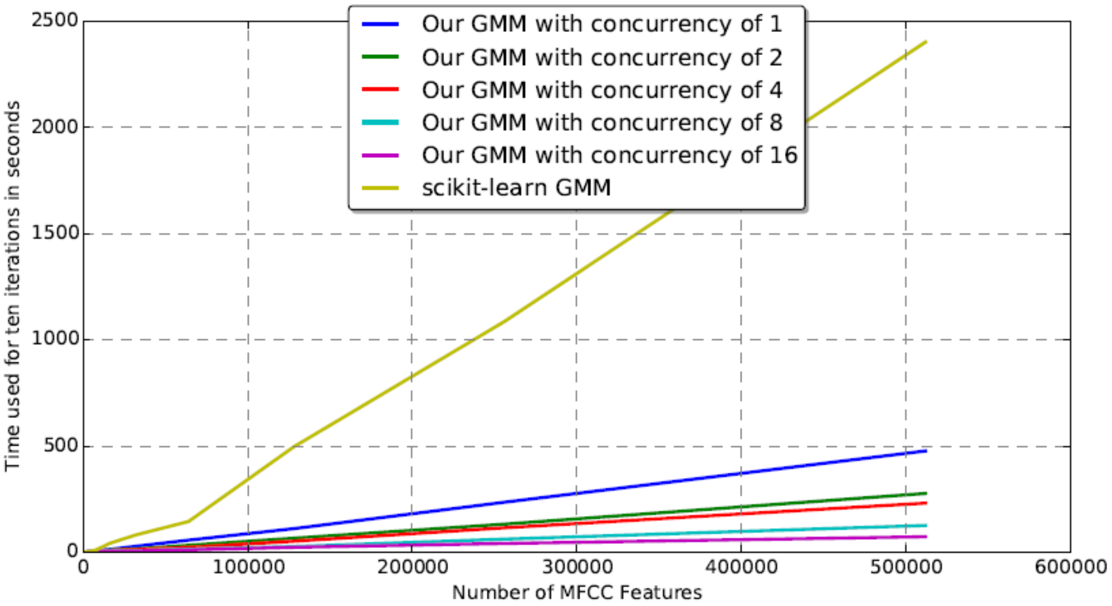


Figure 29: Testing performance using 512,000 MFCC features

We can see from Figure (29) and figure (30), that the proposed GMM performance is outperform the common used GMM from scikit-learn, obviously GMM from scikit-learn consumes a lot of time to process the incremental size of data, unlike our GMM which was time effective when dealing with the same size of incremental data.

The performance of our system is studied using two concurrency cases:

- No concurrency

  When our system processes 512,000 features, our GMM is five times faster than GMM from scikit-learn, therefore we can conclude that in the case of no concurrency our system is clearly faster.

- With concurrency

  The concurrency scalability of the proposed GMM is considered incredibly fast, as the relation between time and the number of the used cores is almost linear, for example when we use 8 cores in our GMM, our GMM will be 19 times faster than GMM from scikit-learn.

5.3 Changing MFCC Parameters

MFCC has several parameters that can affect the accuracy of the proposed system; we conducted tests on some of these parameters to show how these parameters affect the final accuracy of the system. All the tests were

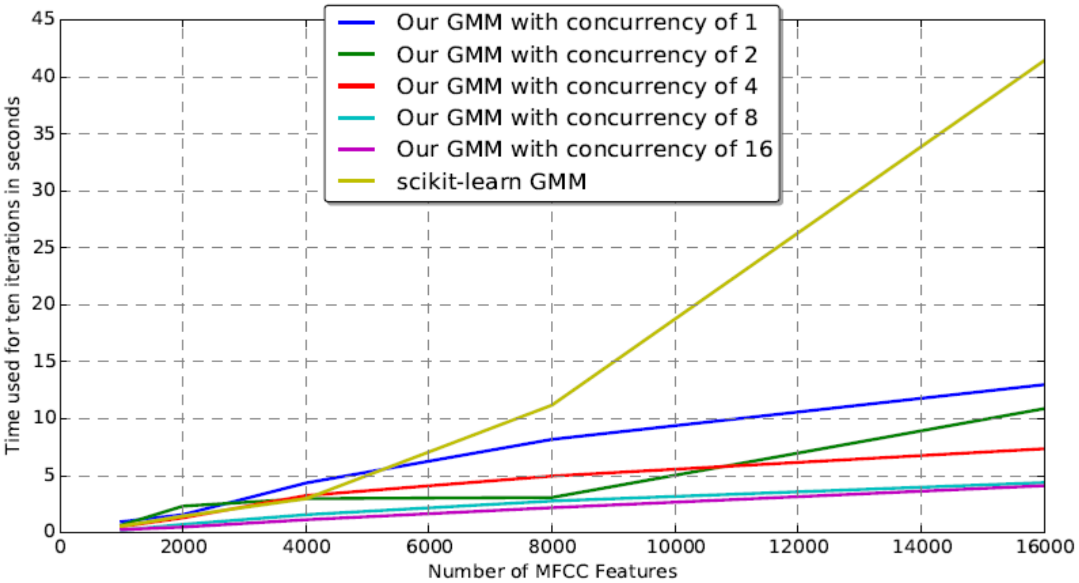


Figure 30: Testing performance using 160,000 MFCC features

performed using 40 reading style voice samples, each voice used with 20 seconds for enrollment and 5 seconds for recognition.

1- Different Number of Cepstrals.

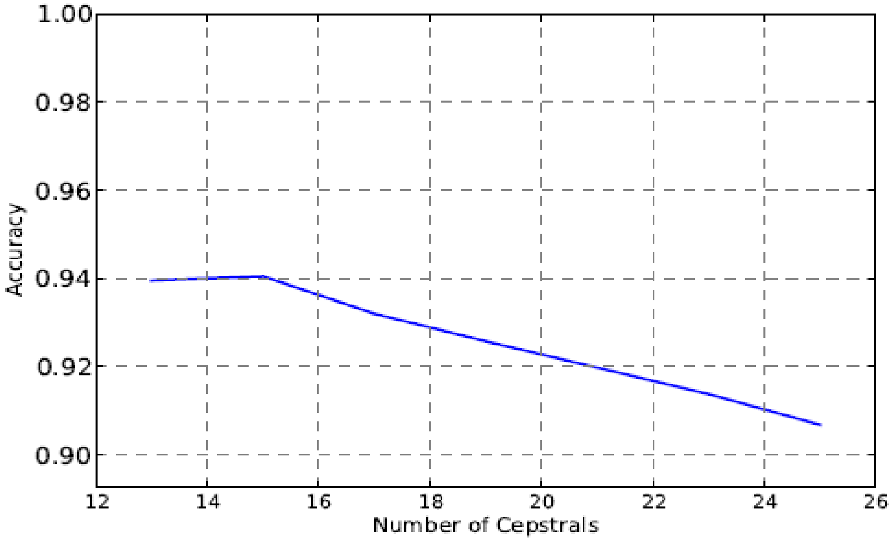


Figure 31: Testing Accuracy with an incremental number of Cepstrals

2- Different Number of Filterbanks

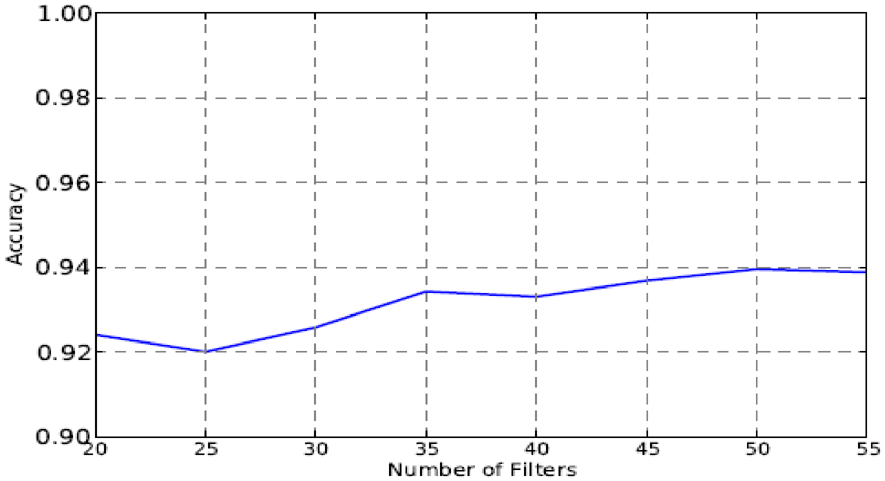


Figure 32: Testing Accuracy with incremental number of filterbanks

In many applications that we reviewed, we found that default value for the filter-banks is 40, in our study we tried to use different values of the filter banks, the accuracy percentage of our trial are displayed in the figure (32). It is obvious from figure (32) that the general trend of the accuracy is incremental when the number of filter banks is increasing.

3- Different Size of Frame

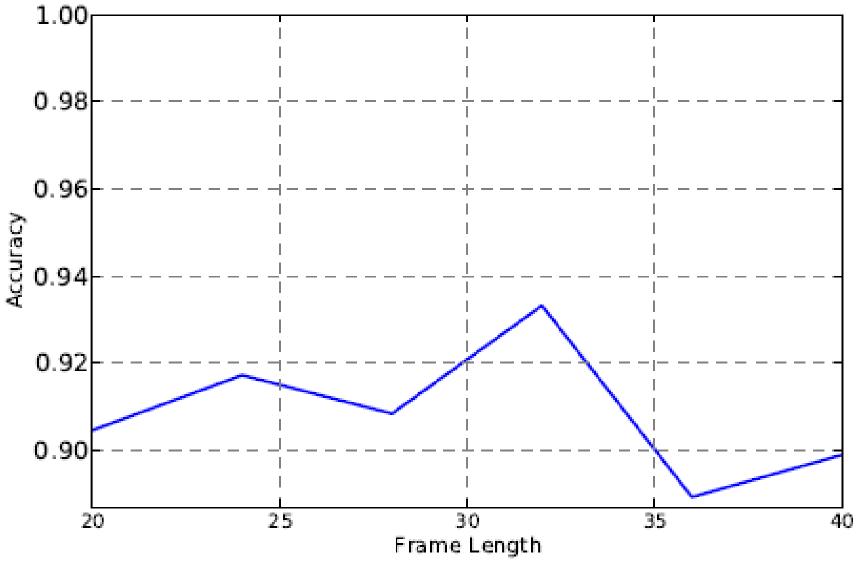


Figure 33: Testing Accuracy with incremental size of frames

Figure (33) shows that the accuracy of the proposed system is decreasing when the frame length exceeds 32ms, this is expectable because the number of features will tremendously increase when the frame length exceeds this point. The huge increment in the features number will increase the ambiguity percentage and decrease the accuracy of the system because of the increasing probability of losing temporal information. However, the accuracy in the worst scenarios as shown in figure (33) will not go less than 88% which is still considered as a high accuracy level.

5.4 Changing LPC Parameters

LPC has several parameters that can affect the accuracy of the proposed system; we conducted tests on some of these parameters to show how these parameters affect the final accuracy of the system. All the tests were performed using 40 reading style voice samples, each voice used with 20 seconds for enrollment and 5 seconds for recognition.

LPC algorithm calculates a signal coefficient and the associated Reflection coefficients.

1. Different Number of Coefficient.

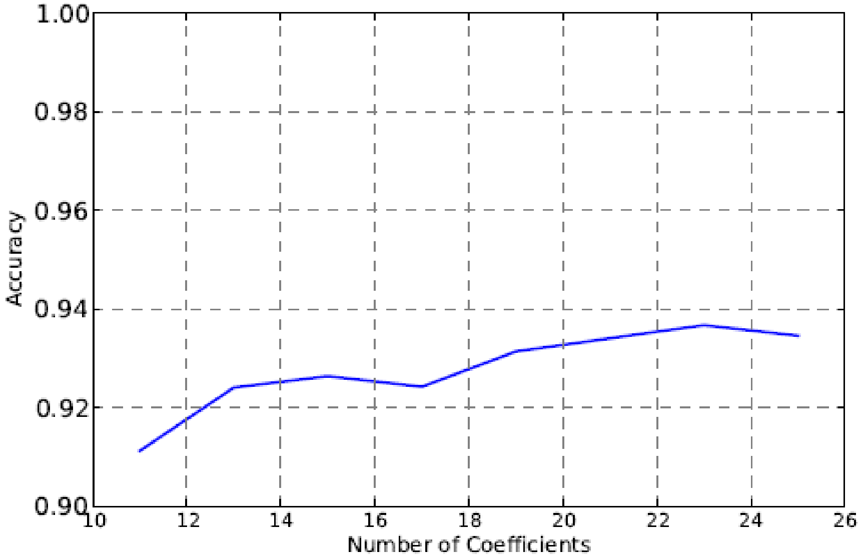


Figure 34: Testing LPCC with incremental number of coefficients

Figure (34) shows that the accuracy increases when the number of coefficients increases, this is because of the increasing information that describes the features of the sampled voice. This will therefore increase the achieved accuracy.

5.5 Changing GMM Components

After conducting several tests on GMM efficiency, specifically tests using different numbers of GMMs components, we found that the accuracy of the system is slightly changed by the changing of the GMM components as shown in figure (35). However, when we used a GMM with a high number of components we noticed that the training process took longer time than the GMMs with less number of components. Therefore we choose to use a GMM with 32 mixtures in our system, this value showed the best performance in all our trials.

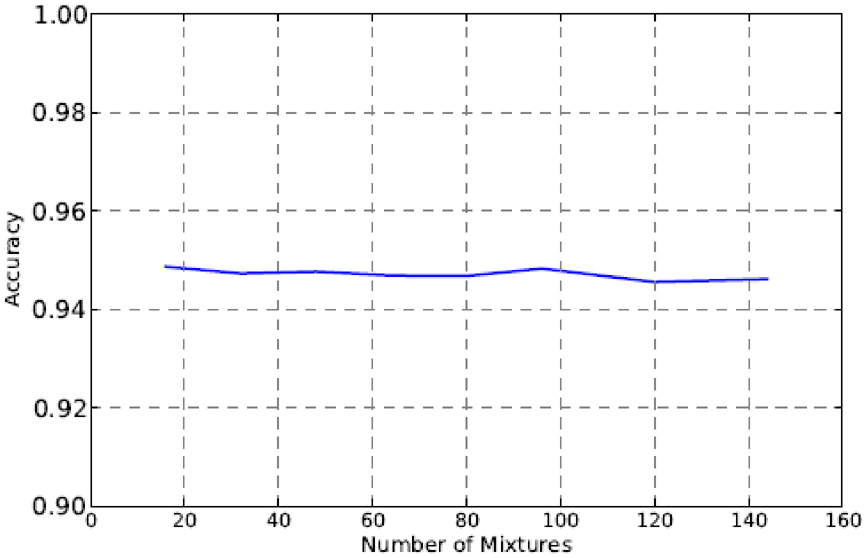


Figure 36: Testing GMM with incremental number of Mixtures

5.6 Different GMM Algorithms

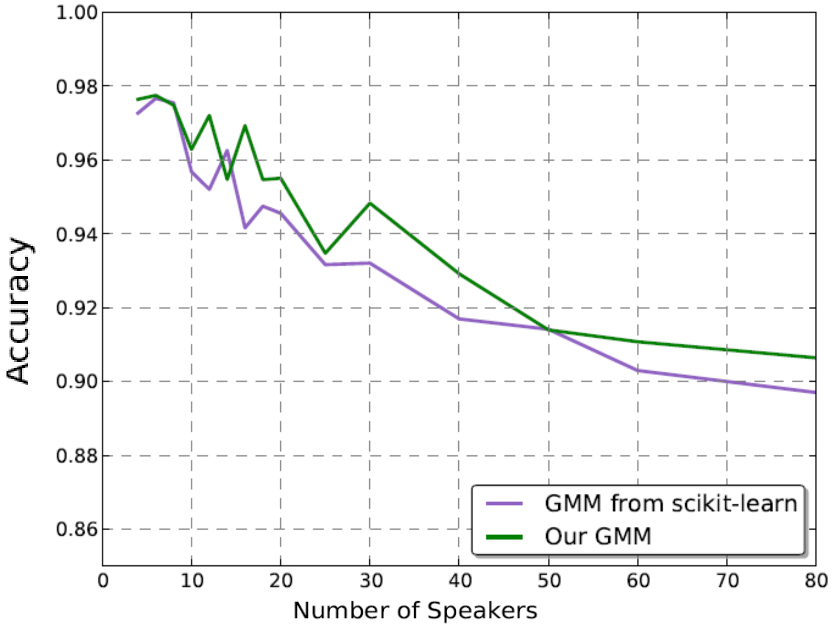


Figure 35: Comparison between scikit-learn GMM and our new GMM

In order to find out the difference in performance between our GMM and scikit-lerarn GMM, we configured the same experiment for both types of GMMs as follows:

- MFCC: 19 ceptrums, frame size 20ms, 40 filterbanks.
- Number of Gaussian Mixtures is set to 32.
- 30s Enrollment utterance, 5s testing utterance.
- 100 test utterance for each speaker.

The results are shown in figure (36)

Figure (36) shows that's the new GMM is performing better than the GMM from scikit-learn. When the number of speakers is small the variation of accuracy levels are considerably high, but when the number of speaker is getting higher the accuracy variation is decreasing. However, in general the proposed GMM is performing better in all the trials performed.

5.7 Accuracy Curve on Different Number of Speakers

It is clear from the previous experiments that when the number of speakers that are loaded to the system is increasing the accuracy of the system is decreasing. Also experiments showed that the duration of the recorded signal in the enrollment stage if it is long, then the accuracy of the speaker recognition system gets higher.

The experiments conducted for estimating the accuracy of the system with various values of the system's parameters is configured as follows:

- "Style-Reading" is the dataset used.
- MFCC: frame size is 32*ms*, 19 cepstrums, 55 filterbanks.
- LPC: frame size is 32*ms*, 15 coefficients.
- GMM from scikit-learn, number of mixtures is 32.
- 20s utterance for enrollment.
- 50 sampled test utterance for each user

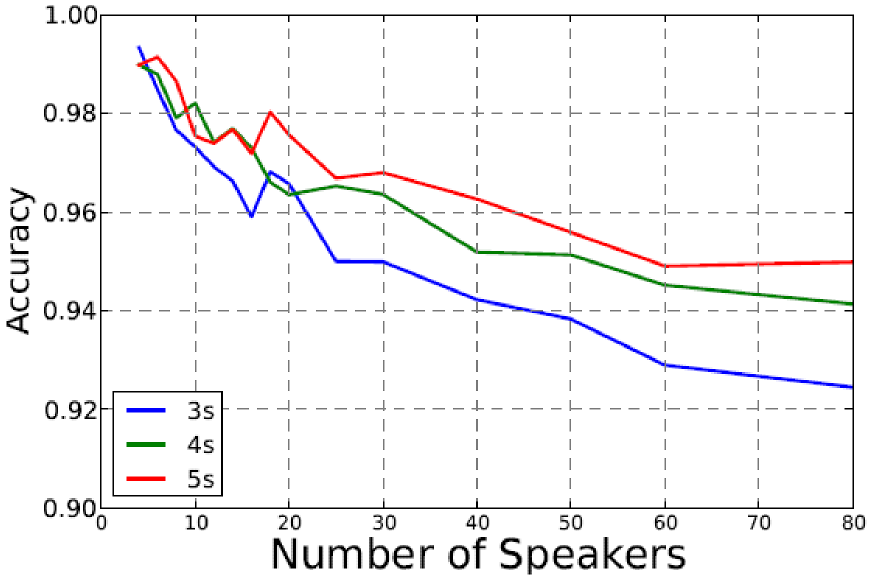


Figure 37: Testing the influence of the number of speakers with different utterances lengths

Experiments were conducted also on different style of datasets. The configuration of this experiment is as followed:

- MFCC: frame size is 32*ms*, 15 cepstrums, 55 filterbanks
- LPC: frame size is 32*ms*, 23 coefficients
- GMM from scikit-learn, number of mixtures is 32
- 20s utterance for enrollment
- 50 sampled test utterance for each user

Figures (37,38,39,40) show the results of the experiment, each point represents an average of 20 independent tests results with randomly selected speakers.

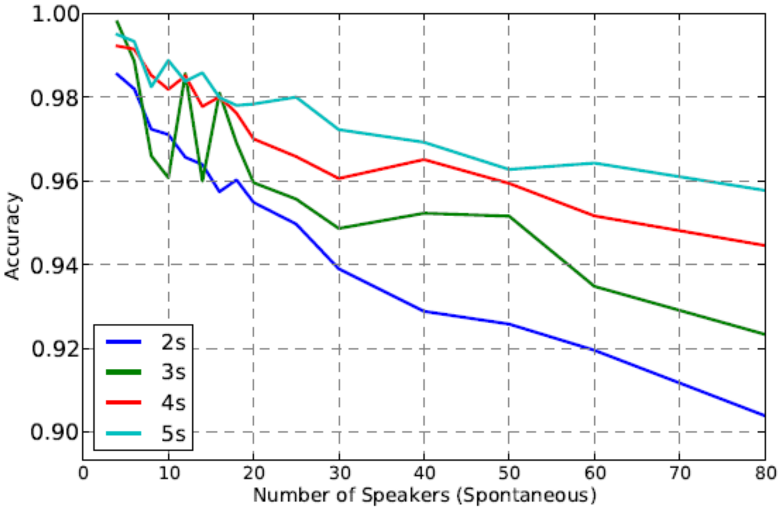


Figure 38: Testing accuracy using spontaneous speaking style

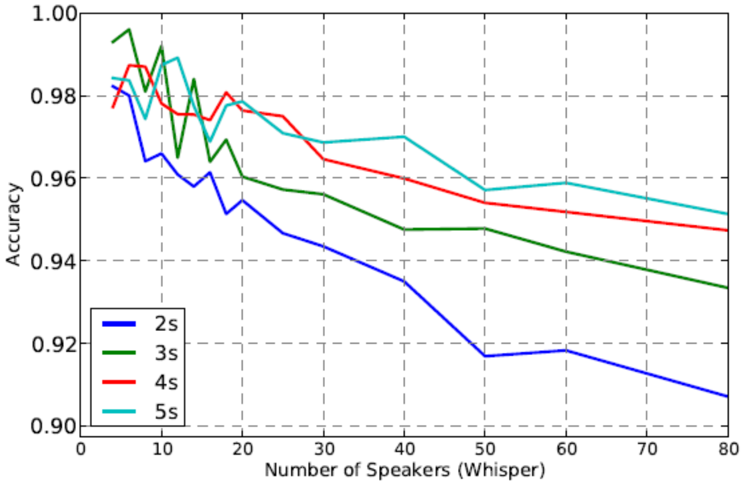


Figure 39: Testing accuracy using Whispering speaking style

5.8 CRBM Performance Test

This experiment showed the effect of number of the speakers on the accuracy of the output of the system using CRBM, CRBM was tested using the following configuration,:

- MFCC: frame size is 32*ms*, 15 cepstrums, 55 filterbanks.
- LPC: frame size is 32*ms*, 23 coefficients.
- CRBM with 32 hidden units.
- 50 sampled test utterance for each user.
- 5s test utterance.

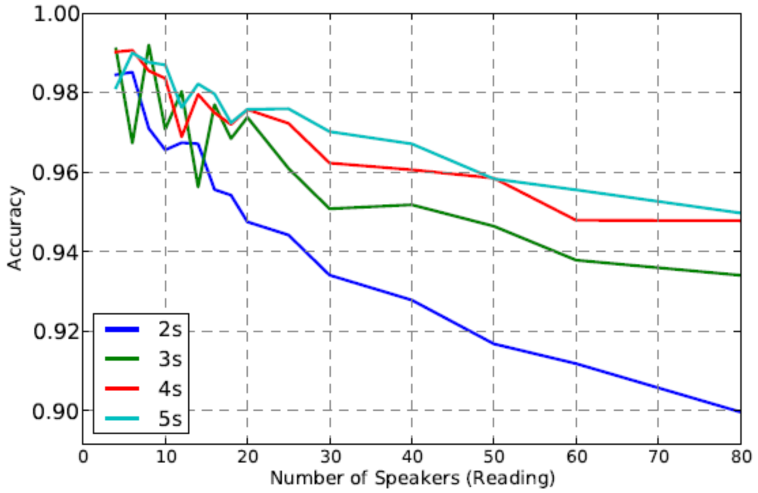


Figure 40: Testing accuracy using reading speaking style

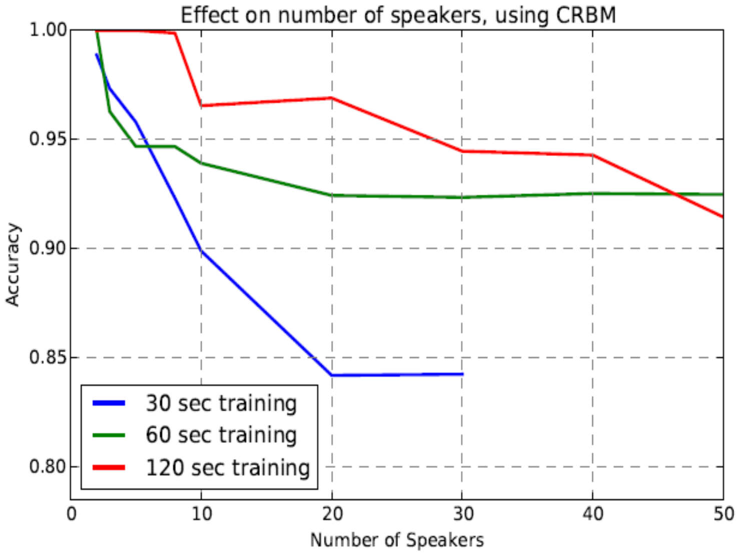


Figure 41: Testing CRBM using different utterances

Figure (41) shows the results from our test using CRBM, it obvious that in order to get an accuracy level like the ones we achieved using GMM's algorithms, the utterance size used for training should be twice the size of the utterance used in GMM's tests.

Chapter 6

Conclusion and Future Work

This chapter summarizes the results and conclusions found after conducting this thesis; it also suggests some ideas that can be carried out in the future as an extension of this thesis scope.

6.1 Conclusion

An analysis of the results generated from the various experiments conducted in this study shows that:

- Using MFCC for extracting features outperform LPCC, LPCC needs double length of utterance to achieve the same results that MFCC achieved**.**
- MFCC is computationally more expensive than LPCC but MFCC is faster and more accurate.
- The new GMM proposed in this study was at least 5 times faster than scikit-learn GMM. Initiating GMM using K-mean II algorithm instead of using random data points is the only but effective difference between scikit-learn GMM and the new GMM used in the proposed system.
- Frame size that is used when splitting the voice signal has a direct effect on the level of accuracy achieved by the system. An increasing frame size leads to lower levels of accuracy because of the loss of temporal information inside the samples in each frame. On the other hand, if the size of the frame is small, this will not provide the system with enough information from the extracted features.
- In all the tests performed on the proposed system, the length of the utterance used for the training phase increases the accuracy of the recognition process when this utterance length is increasing.

- Number of speakers used in the recognition process decreases the accuracy level of the system as long as it is increasing.

6.2 Future work

This study proposed a system that used several features extraction techniques and a variety of modeling algorithms. The following ideas are worthy to be studied in the future:

- Using SVM rather than GMM in the speaker recognition system, and studying the difference in performance between them.
- Using PCA in order to enhance the performance of processing the great number of extracted features.
- Extending the use of the proposed system to be implemented on smart phones.
- Using server- client architecture for designing complicated speaker recognition systems, for example the interface is installed on a smartphone and the processing and computations will be performed on a server that is connected to the related smartphone through internet.
- Preparing the speaker recognition system to be hosted and used online.
- Extending the use of the proposed system to be embedded into real life security systems.

References

Chang, Wen-Wen. "Time Frequency Analysis and Wavelet Transform Tutorial Time-Frequency Analysis for Voiceprint (Speaker) Recognition."

Beigi, Homayoon SM, Stéphane H. Maes, Upendra V. Chaudhari, and Jeffrey S. Sorensen. "IBM model-based and frame-by-frame speaker recognition."*Speaker Recognition and its Commercial and Forensic Applications, Avignon, France* 4 (1998).

Campbell Jr, Joseph P. "Speaker recognition: a tutorial." *Proceedings of the IEEE* 85, no. 9 (1997): 1437-1462.

CHATZARAS, ANARGYROS, and GEORGIOS SAVVIDIS. "Seamless Speaker Recognition." (2015).

Domínguez Sánchez, Carlos. "Speaker Recognition in a handheld computer." (2010).

Dave, Namrata. "Feature extraction methods LPC, PLP and MFCC in speech recognition." *International Journal for Advance Research in Engineering and Technology* 1 (2013).

Tyagi, Vivek, and Christian Wellekens. "On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition." In *ICASSP (1)*, pp. 529-532. 2005.

Mohammed, Eslam Mansour, Mohammed Sharaf Sayed, Abdallaa Mohammed Moselhy, and Abdelaziz Alsayed Abdelnaiem. "LPC and MFCC Performance Evaluation with Artificial Neural Network for Spoken Language Identification." (2013).

Stajano, Frank. "Pico: No more passwords!." In *Security Protocols XIX*, pp. 49-81. Springer Berlin Heidelberg, 2011.

Toader, C., and Frank Stajano. "User authentication for pico: When to unlock a security token." *Master's thesis, University of Cambridge* (2014).

Chowdhury, Md Foezur Rahman, S-A. Selouani, and Douglas O'Shaughnessy. "Text-independent distributed speaker identification and verification using GMM-UBM speaker models for mobile communications." In *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*, pp. 57-60. IEEE, 2010.

Li, Jin-Yu, Bo Liu, Ren-Hua Wang, and Li-Rong Dai. "A complexity reduction of ETSI advanced front-end for DSR." In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 1, pp. I-61. IEEE, 2004.

Brunet, Kevin, Karim Taam, Estelle Cherrier, Ndiaga Faye, and Christophe Rosenberger. "Speaker Recognition for Mobile User Authentication: An Android Solution." In *8ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (SAR SSI)*, p. 10. 2013.

Xu, Min, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. "HMM-based audio keyword generation." In *Advances in Multimedia Information Processing-PCM 2004*, pp. 566-574. Springer Berlin Heidelberg, 2005.

Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." *Applied statistics* (1979): 100-108.

Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027-1035. Society for Industrial and Applied Mathematics, 2007.

Bahmani, Bahman, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. "Scalable k-means++." *Proceedings of the VLDB Endowment* 5, no. 7 (2012): 622-633.

Kenny, Patrick. "Joint factor analysis of speaker and session variability: Theory and algorithms." *CRIM, Montreal,(Report) CRIM-06/08-13* (2005). (Kenny, 2005)

Anjos, André, Laurent El-Shafey, Roy Wallace, Manuel Günther, Christopher McCool, and Sébastien Marcel. "Bob: a free signal processing and machine learning toolbox for researchers." In *Proceedings of the 20th ACM international conference on Multimedia*, pp. 1449-1452. ACM, 2012.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12 (2011): 2825-2830.

2015. http://www.behaviosec.com/wp-content/uploads/2012/01/BehavioSec-Behaviometrics.pdf.

2015. http:///github.com/amaurycrickx/recognito.

2015. http://mistral.univ-avignon.fr/ index en.html.

[http://speech.fit.vutbr.cz];, all:. 2015. 'Joint Factor Analysis Matlab Demo | Speech Processing Group'.*Speech.Fit.Vutbr.Cz*. http://speech.fit.vutbr.cz/software/joint-factor-analysis-matlab-demo.

Beigi, Homayoon. *Fundamentals of speaker recognition*. Springer Science & Business Media, 2011.

Cmusphinx.sourceforge.net,. 2015. 'Tutorialsphinx4_Doc_Sphinx4-Faq._Html [Cmusphinx Wiki]'. http://cmusphinx.sourceforge.net/wiki/tutorialsphinx4/doc/Sphinx4-faq. html#speaker identification.

Code.google.com,. 2015. 'Voiceid - Speaker Recognition/Identification System In Python - Google Project Hosting'. http://code.google.com/p/voiceid/.

Flanagan, James L. 1972. *Speech Analysis; Synthesis And Perception*. Berlin: Springer-Verlag.

GitHub,. 2015. 'Sikoried/Jstk'. https://code.google.com/p/jstk/.

Kenny, Patrick, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. 2007. 'Joint Factor Analysis Versus Eigenchannels In Speaker Recognition'. *IEEE Transactions On Audio, Speech And Language Processing* 15 (4): 1435-1447. doi:10.1109/tasl.2006.881693.

Miller, R. L. 1959. 'Nature Of The Vocal Cord Wave'. *The Journal Of The Acoustical Society Of America* 31 (6): 667-677. doi:10.1121/1.1907771.

Numpy.org,. 2015. 'Numpy — Numpy'. http://www.numpy.org/.

People.csail.mit.edu,. 2015. 'Pyaudio: Portaudio V19 Python Bindings'. http://people.csail.mit.edu/hubert/pyaudio/.

Pypi.python.org,. 2015. 'Bob.Spear 1.1.8 : Python Package Index'. https://pypi.python.org/pypi/bob.spear.

Pypi.python.org,. 2015. 'Pyssp 0.1.6.5 : Python Package Index'. https://pypi.python.org/pypi/pyssp.

Pypr.sourceforge.net,. 2015. 'Welcome To Pypr'S Documentation! — Pypr V0.1Rc3 Documentation'. http://pypr.sourceforge.net/.

PyQt,. 2015. 'Pyqt'. *Sourceforge*. http://sourceforge.net/projects/pyqt/.

Python.org,. 2015. 'Welcome To Python.Org'. https://www.python.org/.

Ramírez, Javier, José C Segura, Carmen Benítez, Ángel de la Torre, and Antonio Rubio. 2004. 'Efficient Voice Activity Detection Algorithms Using Long-Term Speech Information'. *Speech Communication* 42 (3-4): 271-287. doi:10.1016/j.specom.2003.10.002.

Repository.voxforge1.org,. 2015. 'Voxforge Repository'. http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/Audio/Main/8kHz_16bit/.

Reynolds, D.A., and R.C. Rose. 1995. 'Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models'. *IEEE Transactions On Speech And Audio Processing* 3 (1): 72-83. doi:10.1109/89.365379.

Reynolds, Douglas A., Thomas F. Quatieri, and Robert B. Dunn. 2000. 'Speaker Verification Using Adapted Gaussian Mixture Models'. *Digital Signal Processing* 10 (1-3): 19-41. doi:10.1006/dspr.1999.0361.

Scikit-learn.org,. 2015. 'Scikit-Learn: Machine Learning In Python — Scikit-Learn 0.16.1 Documentation'. http://scikit-learn.org/.

Scikits.appspot.com,. 2015. 'Scikits - Scikits.Talkbox'. http://scikits.appspot.com/talkbox.

Scipy.org,. 2015. 'Scipy.Org — Scipy.Org'. http://www.scipy.org/.

Shrawankar, Urmila, and V M Thakare. 2013. 'Techniques For Feature Extraction In Speech Recognition System : A Comparative Study'. *Arxiv.Org*. http://arxiv.org/abs/1305.1145.

Sox.sourceforge.net,. 2015. 'Sox - Sound Exchange | Homepage'. http://sox.sourceforge.net/.

Wikipedia,. 2015. 'Restricted Boltzmann Machine'. http://en.wikipedia.org/wiki/Restricted_Boltzmann_machine#cite_note-1.

Wikipedia,. 2015. 'Mel-Frequency Cepstrum'. http://en.wikipedia.org/wiki/Mel-frequency_cepstrum.

Doc.madlib.net,. 2015. 'Madlib: K-Means Clustering'. http://doc.madlib.net/master/group__grp__kmeans.html.

El Hannani, Asmaa, Dijana Petrovska-Delacrétaz, Benoît Fauve, Aurélien Mayoue, John Mason, Jean-François Bonastre, and Gérard Chollet. "Text-independent speaker verification." In *Guide to Biometric Reference Systems and Performance Evaluation*, pp. 167-211. Springer London, 2009.

Alam, Md Jahangir, Tomi Kinnunen, Patrick Kenny, Pierre Ouellet, and Douglas O'Shaughnessy. "Multitaper MFCC and PLP features for speaker verification using i-vectors." *Speech Communication* 55, no. 2 (2013): 237-251

Julio César Martínez-Romo, Francisco Javier Luna-Rosas, Miguel Mora-González, Carlos Alejandro de Luna-Ortega and Valentín López-Rivas (2012). Optimal Feature Generation with Genetic Algorithms and FLDR in a Restricted-Vocabulary Speech Recognition System, Bio-Inspired Computational Algorithms and Their Applications, Dr. Shangce Gao (Ed.), ISBN: 978-953-51-0214-4, InTech, DOI: 10.5772/36135.

Kinnunen, Tomi, and Haizhou Li. 2010. 'An Overview Of Text-Independent Speaker Recognition: From Features To Supervectors'. *Speech Communication* 52 (1): 12-40. doi:10.1016/j.specom.2009.08.009.

Feng, Ling. "Speaker recognition." PhD diss., Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2004.

Ahmed, Nasir, T. Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." *Computers, IEEE Transactions on* 100, no. 1 (1974): 90-93.

Reynolds, Douglas A. "An overview of automatic speaker recognition." In*Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)(S. 4072-4075)*. 2002.

Ee.columbia.edu,. 2015. 'Contents'. http://www.ee.columbia.edu/ln/LabROSA/doc/HTKBook21/node1.html.

Bilmes, Jeff A. "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models."*International Computer Science Institute* 4, no. 510 (1998): 126.

Winograd, Shmuel. "On computing the discrete Fourier transform."*Mathematics of computation* 32, no. 141 (1978): 175-199.